Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Electric Drives and Traction

MATHEMATICAL MODEL AND VALIDATION OF POWERTRAIN

by

Bc. JAN KACETL

A master thesis submitted to
the Faculty of Electrical Engineering, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Master.

Master degree study programme: Electrical Machines, Apparatus and Drives

Prague, January 2016

Supervisor:
Ing. TOMÁŠ HAUBERT
Department of Electric Drives and Traction
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
166 27 Prague 6
Czech Republic

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra elektrických pohonů a trakce

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Jan Kacetl**

Studijní program: Elektrotechnika, energetika a management
Obor: Elektrické stroje, přístroje a pohony

Název tématu: **Matematický model elektrického vozidla a jeho validace na pracovišti VTP Roztoky**

Pokyny pro vypracování:

1) V prostředí MATLAB/Simulink vytvořte matematický model pohonné jednotky elektrického vozidla, která se nachází na výzkumném pracovišti VTP Roztoky.
2) Dále vytvořte model baterií a identifikujte parametry motoru a bateriového článku. Navržený model porovnejte s reálným pohonem z hlediska účinnosti v různých provozních stavech.
3) Tento model implementujte do systému dSpace DS1103 a navrhněte software pro řízení dynamometru, měniče a nadřazeného PC. Pro výzkumné pracoviště VTP Roztoky dále navrhněte a implementujte do systému dSpace DS1103 bezpečnostní koncept měření elektrických pohonů.

Seznam odborné literatury:

[1] MOHAN, Ned.: Advanced electric drives: analysis, control, and modeling using MATLAB/Simulink. Hoboken: Wiley, 2014
[2] JAZAR, Reza N.: Vehicle dynamics: theory and application. 2nd ed. New York: Springer, 2014
[3] Pera, M.-C., Hissel, D., and Gualous, H.: Electrochemical Components. Somerset, NJ, USA: John Wiley & Sons, 2013.

Vedoucí: Ing. Tomáš Haubert

Platnost zadání: do konce letního semestru 2016/2017

L.S.

Ing. Jan Bauer, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 1. 4. 2016

Anotace

Tato diplomová práce se zabývá matematickým popisem, modelováním a identifikací parametrů pohonné jednotky elektrického automobilu, která se nachází ve VTP Roztoky. Tato pohonná jednotka se skládá z bateriového zdroje, invertoru řízeného metodou DTC a asynchronního motoru. Pro každou z těchto částí je zde matematický model a metoda identifikace parametrů modelu. Výsledky simulací těchto modelů jsou pak validovány pomocí měření na reálném zařízení. V této diplomové práci je dále vyvinut a představen řídící systém simulačního zařízení pro simulaci jízdy elektrického vozidla předepsanou rychlostí po trase s daným výškovým profilem. Během této hardwarové simulace jsou online počítané jízdní odpory, vzhledem k parametrům trasy a rychlosti vozidla, které jsou pak aplikovány pomocí řízeného dynamometru spojeného hřídelí s pohonným motorem. Tento řídící algoritmus v sobě zahrnuje bezpečnostní prvky zajišťující bezpečný chod soustrojí. Řídící algoritmus je implementován do systému dSpace DS1103. Komunikace mezi jednotlivými prvky simulačního systému je realizován pomocí komunikačních potokolů CAN a RS232.

Klíčová slova:

asynchronní motor, invertor, baterie, pohonná jednotka, model, identifikace, účinnostní mapa, dSpace, Matlab/Simulink, stateflow, CAN, CANOpen, RS232, EV, jízda, řídící systém, měřící stolice jízdní odpory.

Abstract

This master thesis deals with the mathematical description, modelling and system identification of EV powertrain test bench at VTP Roztoky. The powertrain consists of a battery pack, a power inverter using DTC and an induction motor. For each of these components, there is a mathematical description, the methodology of system identification and a model validation by comparing simulation results with real device measurement. This master thesis also discusses the control algorithm of the test bench to simulate the EV drive along a track with known altitude by the defined speed of the vehicle in respect to drive resistance forces. The drive resistance forces are simulated by a controlled dynamometer. The control algorithm includes safety features to avoid test bench failure and is implemented in dSpace DS1103. The communication between actuators is established by the CAN communication protocol and the RS232 communication protocol.

Keywords:

induction motor, power inverter, battery, powertrain, model, identification, efficiency map, dSpace, Matlab/Simulink, stateflow, CAN, CANOpen, RS232, EV, drive, control system, test bench, drive resistance forces

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečnch prací.

V Praze dne 27. května .....

Acknowledgement

First of all, I would like to express my gratitude to my master thesis supervisor, Ing. Tomáš Haubert. He has been a constant source of encouragement and insight during my research and helped me with numerous problems and professional advancements. If there are any shortcomings, then they are all mine.

Special thanks go to Ing. Vít Hlinovský, CSc. for the continuous support of my Bachelor and Master studies.

Finally, my greatest thanks go to my family members, for their infinite patience and care.

"Nothing would be done at all if a man waited until he could do it so well that no one could find fault with it"

— John Henry Newman

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the Master thesis is introduced. It starts with motivation, followed by Problem Statement. The introduction ends by the Goals of the Master Thesis and the Structure of the Master Thesis.

1.1 Motivation

This master thesis builds upon the project of Josef Bozek Competence Centre for Automotive Industry, supported by Technology agency of the Czech Republic. The project focuses on the research and optimization of electrical and hybrid propulsion. The research investigates possibilities of energy consumption reduction of the EV. One possible way is to adapt the vehicle speed to the track parameters. To get the speed distribution along the track, called a speed profile, a mathematical optimization algorithm is used. The algorithm requires a mathematical model of the EV in respect to the vehicle dynamics and powertrain energy conversion efficiency.

The laboratory at VTP Roztoky is equipped by a test bench. A control system, controlling the test bench in respect to the mathematical model of the EV, could give us a possibility to validate the optimization algorithm result by real device measurement.

1.2 Problem Statement

The EV powertrain consists of a battery pack, a power converter and an electrical motor. A mathematical model of the powertrain gives insights into electromechanical energy conversion, and enables improvement of the process.

To simulate a real system, parameters of the system have to be identified first. A test bench is a real device simulation of the powertrain. It is used to measure the powertrain in different operating states, defined by the motor speed and torque. The load torque is generated by a dynamometer. The powertrain of the EV goes through various operating states during the drive. To simulate an EV drive along a certain track, proper load torque

is necessary to be applied at every moment of the drive. It follows that, the control system of the test bench has to control both the dynamometer load torque and the drive motor speed, where the load torque is based on the EV model and the track parameters.

1.3 Goals of the Master Thesis

There are two main goals of this Master Thesis:
The first is to create a mathematical model of the EV powertrain.
The second one is to develop a simulation system for EV drive simulation by modification of the standard test bench.
The steps to achieve these goals are:

1. Mathematical description of the powertrain.
2. Identification of the system parameters.
3. Model validation.
4. Test bench modification.
5. Development of test bench control system.
6. Safety features implementation.

1.4 Structure of the Master Thesis

The thesis is organized into 5 chapters as follows:

1. *Introduction*: Describes the motivation behind our efforts together with our goals.
2. *Mathematical models*: Describes the mathematical models of components of the EV powertrain. Further, the system identification methods are discussed and applied on real devices.
3. *Simulation system*: Discusses development and function of the control system of the test bench in VTP Roztoky.
4. *Results*: Evaluates the achieved results of the master thesis.
5. *Conclusion*: Summarizes the results of our research, suggests possible subjects for further research, and concludes the thesis.

Chapter 2

Mathematical model

This chapter discusses the mathematical model of the EV powertrain. In the following sections, the mathematical models of the induction motor, the power inverter, the Direct Torque Control (DTC) and the battery pack are described. Moreover, these sections include the system identification methods applied on real devices. The simulation results are then compared with the results of the real device measurements. The models are set in the following order: Induction motor, Power inverter, DTC, Battery pack.

2.1 Induction motor

The induction motor is an AC electric machine converting electrical power into mechanical power. Stator of the motor is made of electrical steel sheets to minimize hysteresis and eddy current losses. The stator windings are distributed sinusoidally around the stator and are placed in slots. The rotor is also made of electrical steel sheets. The rotor winding can be wounded or made as a squirrel-cage. The air gap between stator and rotor must be as small as possible to minimize leakage flux. The torque is produced by interaction between the magnetic flux produced by the stator windings and the current of the rotor windings. The rotor current is induced in the rotor winding by the rotating magnetic field in the air gap. Thus the induction motor is also known as a rotating transformer.

2.1.1 Mathematical model of induction motor

The construction of the three-phase induction motor with squirrel−cage is simple as mentioned in the introduction of this section. On the other hand, mathematical description of the system is quite complex. To simplify the mathematical description, the following conditions are defined:

- Symmetrical stator windings

- Symmetrical rotor windings
- Invariable air gap
- Sinusoidal distribution of windings
- Core losses are not taken into account
- Magnetic material is operated in its linear region and has infinite permeability

To describe a quantity in the three-phase system, a space vector is used. Unlike phasors, space vectors are also applicable under dynamic condition [5]. The space vector can easily describe the field quantities distributed sinusoidally in the air gap. The resultant voltage, current and flux linkage space vectors are calculated as sums of products of instantaneous phase values and the winding orientations of every single phase. To simplify the system, mathematical transformation of the system is used as described in the following section.

2.1.1.1 Mathematical transformation

A space vector of a physical quantity in three-phase system, under condition of sinusoidal distribution of winding, is described by equation 2.1.

$$\overrightarrow{i_s}(t) = K(i_a(t)e^{0j} + i_b(t)e^{\frac{2\pi}{3}j} + i_c(t)e^{\frac{4\pi}{3}j}) \tag{2.1}$$

Equation 2.2 is valid for three-phase wye connected power system.

$$i_a(t) + i_b(t) + i_c(t) = 0 \tag{2.2}$$

Substituting equation 2.2 into equation 2.1 results in equation 2.3.

$$\overrightarrow{i_s}(t) = K(\frac{3}{2}i_a(t) + j\frac{\sqrt{3}}{2}(i_a(t) + 2i_b(t))) \tag{2.3}$$

If we separate real and complex parts of equation 2.3 and rewrite equation 2.3 in a matrix form with an index α for the real part and β for the imaginary part we get equation 2.4. This transformation is called Clarke transformation, or α, β transformation. The transformation is shown in figure 2.1.

$$\begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} = K \begin{bmatrix} \frac{3}{2} & 0 & 0 \\ \sqrt{3} & \frac{\sqrt{3}}{2} & 0 \end{bmatrix} \begin{bmatrix} i_a(t) \\ i_b(t) \\ i_c(t) \end{bmatrix} \tag{2.4}$$

The α, β is a reference frame fixed with the stator. In general, α, β reference frame can be transformed into rotating reference frame rotating by relative angular speed ω, as shown in figure 2.2.

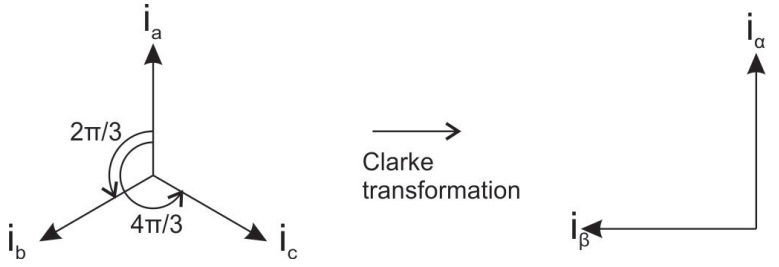


Figure 2.1: Clarke transformation

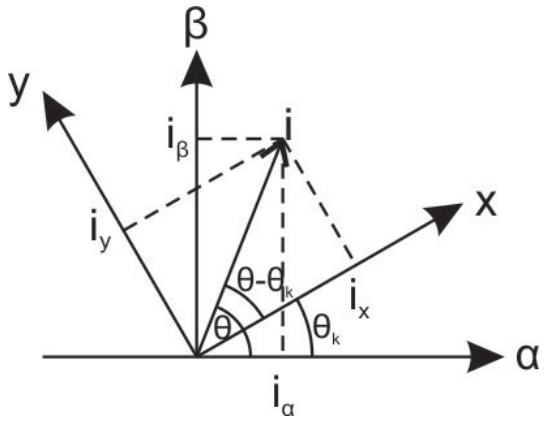


Figure 2.2: Park's transformation

The transformation from α, β reference frame into rotating reference frame is described by equation 2.5.

$$\begin{bmatrix} i_x(t) \\ i_y(t) \end{bmatrix} = \begin{bmatrix} cos(\Theta) & sin(\Theta) \\ -sin(\Theta) & cos(\Theta) \end{bmatrix} \begin{bmatrix} i_\alpha(t) \\ i_\beta(t) \end{bmatrix} \tag{2.5}$$

The relation between a space vector of a quantity $\overrightarrow{x}^k$ in the reference frame rotating by angular speed ω_k and space vector of the same quantity in the α, β reference frame $\overrightarrow{x}^s$ are expressed as follows in equation 2.6.

$$\overrightarrow{x}^k = \overrightarrow{x}^s e^{-j\vartheta_k} \tag{2.6}$$

2.1.2 Circuit equations

Using space vector, the circuit equations of stator and rotor circuits are described by equations 2.7 and 2.8 according to Kirchhoff's circuit law.

$$\overrightarrow{u}^k_s e^{j\vartheta_k} = R_s \overrightarrow{i}^k_s e^{j\vartheta_k} + \frac{d}{dt}(\overrightarrow{\psi}^k_s e^{j\vartheta_k}) \tag{2.7}$$

$$\overrightarrow{u}^k_r e^{j(\vartheta_k - \vartheta)} = R_r \overrightarrow{i}^k_r e^{j(\vartheta_k - \vartheta)} + \frac{d}{dt}(\overrightarrow{\psi}^k_r e^{j(\vartheta_k - \vartheta)}) \tag{2.8}$$

Where $\overrightarrow{u}_s^k$ is the space vector of stator voltage, R_s is the stator phase resistance, $\overrightarrow{i}_s^k$ is the space vector of stator current and $\overrightarrow{\psi}_s^k$ is the space vector of stator flux linkage. The same principle in equation 2.7 applies to the rotor equation 2.8. The index k denotes a general reference frame rotating by angular speed ω_k.

By differentiating the last elements in both equations we get equations 2.9 and 2.10.

$$\overrightarrow{u}_s^k = R_s \overrightarrow{i}_s^k + \frac{d\overrightarrow{\psi}_s^k}{dt} + j\omega_k \overrightarrow{\psi}_s^k \tag{2.9}$$

$$\overrightarrow{u}_r^k = R_r \overrightarrow{i}_r^k + \frac{d\overrightarrow{\psi}_r^k}{dt} + j(\omega_k - \omega) \overrightarrow{\psi}_r^k \tag{2.10}$$

The stator and rotor flux linkages are expressed by space vector equations 2.11 and 2.12.

$$\overrightarrow{\psi}_s^k = L_s \overrightarrow{i}_s^k + L_m \overrightarrow{i}_r^k \tag{2.11}$$

$$\overrightarrow{\psi}_r^k = L_m \overrightarrow{i}_s^k + L_r \overrightarrow{i}_r^k \tag{2.12}$$

These equations describe stator and rotor circuit in a general reference frame rotating by rotation speed ω_k. Some of the frequently used reference frames and their rotation speeds are listed in table 2.1.

Reference Frame	**Marking**	**Rotation speed**
Reference frame fixed with stator	α, β	$\omega_k = 0$
Reference frame fixed with rotor	k, l	$\omega_k = \omega$
Reference frame fixed with rotor magnetic flux linkage	d, q	$\omega_k = \omega_1$

Table 2.1: Reference frame overview

In our model, the reference frame fixed with the rotor magnetic flux linkage is used. The circuit equations in the d, q reference frame are expressed in matrix form as shown in 2.13 and 2.14.

$$\begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix} = R_s \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix} + \omega_s \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \end{bmatrix} \tag{2.13}$$

$$\begin{bmatrix} u_{rd} \\ u_{rq} \end{bmatrix} = R_r \begin{bmatrix} i_{rd} \\ i_{rq} \end{bmatrix} + \frac{d}{dt} \begin{bmatrix} \psi_{rd} \\ \psi_{rq} \end{bmatrix} + \omega_{sl} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \psi_{rd} \\ \psi_{rq} \end{bmatrix} \tag{2.14}$$

Where the indexes d and q represent projections of the space vector to the d and q axis of the reference frame.

The stator and rotor magnetic flux linkage equations in d, q reference frame are expressed in matrix form as shown in 2.15.

$$[\psi_{dq}] = [M]\,[i_{dq}] \tag{2.15}$$

Where the $[\psi_{dq}]$ is the vector of stator and rotor magnetic flux linkage 2.16,

$$[\psi_{dq}] = \begin{bmatrix} \psi_{sd} \\ \psi_{sq} \\ \psi_{rd} \\ \psi_{rq} \end{bmatrix} \tag{2.16}$$

$[M]$ is the matrix of Self and Mutual inductances defined in 2.17

$$[M] = \begin{bmatrix} L_s & 0 & L_m & 0 \\ 0 & L_s & 0 & L_m \\ L_m & 0 & L_r & 0 \\ 0 & L_m & 0 & L_r \end{bmatrix} \tag{2.17}$$

and $[i_{sq}]$ is the vector of stator and rotor currents as described in 2.18.

$$[i_{sq}] = \begin{bmatrix} i_{sd} \\ i_{sq} \\ i_{rd} \\ i_{rq} \end{bmatrix} \tag{2.18}$$

Stator and rotor currents are expressed from the equation 2.15 by left multiplication of the equation by inverse matrix of Self and Mutual inductances. This results in equation 2.19.

$$[i_{dq}] = [M]^{-1} [\psi_{dq}] \tag{2.19}$$

If the rotor is a squirrel-cage type, then rotor voltage is equal to zero as described in 2.20.

$$\begin{bmatrix} u_{rd} \\ u_{rq} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.20}$$

As space vector of rotor magnetic flux linkage is aligned with d axis of d, q reference frame, ψ_{rq} is considered as $\psi_{rq} = 0$ and $\frac{d\psi_{rq}}{dt} = 0$.

The torque of the induction motor is the sum of torques produced by windings of both axis d and q. The torque produced by d-axis winding is applied in counter-clockwise direction and is considered as positive value. This torque is expressed in equation 2.21.

$$M_{rd} = \frac{3}{2} p \psi_{rq} i_{rd} \tag{2.21}$$

Similarly, the torque produced by q-axis winding is applied in clockwise direction and is considered as negative. This torque is expressed in equation 2.22.

$$M_{rq} = -\frac{3}{2} p \psi_{rd} i_{rq} \tag{2.22}$$

The resulting torque is the sum of these two torques as described in equation 2.23.

$$M_{em} = M_{rd} + M_{rq} \tag{2.23}$$

By substitution of equations 2.21 and 2.22 into 2.23 we get equation 2.24.

$$M_{em} = \frac{3}{2} p(\psi_{rq} i_{rd} - \psi_{rd} i_{rq}) \tag{2.24}$$

By another substitution of equation 2.15 into equation 2.24 for ψ_{rd} and ψ_{rq} we get equation 2.25 [6].

$$M_{em} = \frac{3}{2} p L_m (i_{sq} i_{rd} - i_{sd} i_{rq}) \tag{2.25}$$

The relation between electrical and mechanical angular velocity is shown in equation 2.26.

$$\omega_{mech} = p\omega_{el} \tag{2.26}$$

The last equation of the equation system is the equation of motion, 2.27 [7][5][8].

$$M_{el} - M_L = J \frac{d\omega_{mech}}{dt} \tag{2.27}$$

2.1.3 System Identification

The equations describing the system of induction motor in the previous section have parameters which are necessary to be identified to fit the model to the induction motor used in our system. These parameters and their designation are shown in table 2.2.

Designation	Description
R_s	Stator resistance
$L_{s\sigma}$	Stator leakage inductance
R_r	Rotor resistance
$L_{r\sigma}$	Rotor leakage inductance
L_m	Mutual inductance
J	Combined inertia of the motor and the load
p	Number of pole pairs

Table 2.2: Induction motor parameters overview

2.1.3.1 Number of pole pairs

The number of pole pairs is always a part of the label of the induction motor. In this system a four-pole induction motor ($p = 2$) is used.

2.1.3.2 Combined inertia

In this part, only the inertia of the test bench set is taken into account. The inertia of the simulated EV is discussed in section 3.4.1. The combined inertia of the induction motor and the load are either calculated or measured indirectly. The test bench set consists of the induction motor and the dynamometer. The value of the inertia of the dynamometer is known from the label of the machine $J_L = 0.445kg \cdot m^2$. The inertia of the induction motor is necessary to be established. The calculation of the inertia of a rotating rigid body is based on investigation of the rotational energy, which leads to equation 2.28.

$$J = \int_{(m)} r^2 dm \tag{2.28}$$

Where dm is an elemental point mass and r is the distance of the element from the axis of rotation. A squirrel cage rotor is a non-homogeneous cylinder as shown in figure 2.3.

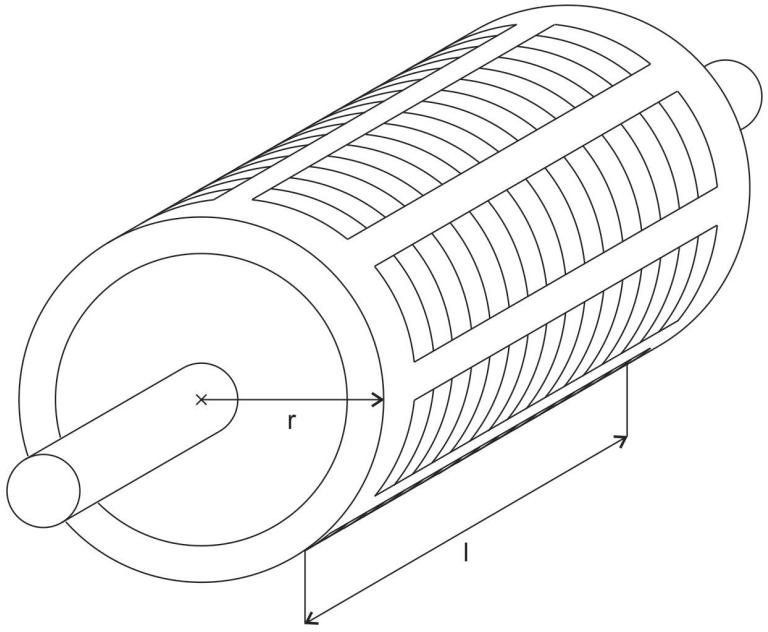


Figure 2.3: Rotor geometry

Therefore, the calculation of the inertia is not trivial. Moreover, the geometry of the rotor of the induction motor used in the system is unknown so the inertia of the motor must be calculated from the deceleration curve as shown in figure 2.4.

This calculation comes from equation 2.27. Replacing the velocity derivative by difference, the inertia can be expressed as shown in equation 2.29.

$$J = \frac{T_2 - T_1}{n_2 - n_1} \frac{60}{2\pi} M_f \tag{2.29}$$

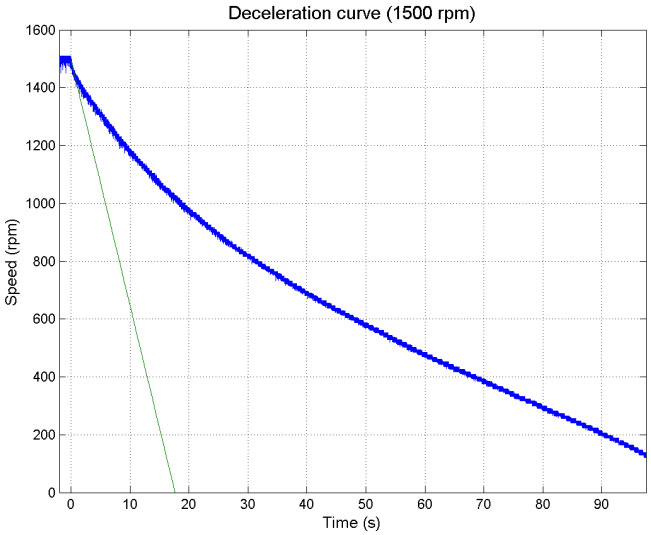


Figure 2.4: Deceleration curve measurement

Where M_f is the torque produced by mechanical friction calculated from the mechanical losses. The mechanical losses come from no-load test. The inertia is established as $J = 0.043kg \cdot m^2$, from the deceleration curve in figure 2.4.

2.1.3.3 Stator resistance

The stator resistance R_s corresponds to the stator phase resistance. The resistance measurement is based on Ohm's Law. The stator winding is supplied by a DC current and the resulting DC voltage is measured, as shown in figure 2.5. The resistance is measured at $20°C$. The supporting current must be small enough so as not to warm up the stator winding. If the temperature is different from $20°C$, R_{x20} must be calculated.

$$R_x = \frac{U_x}{I_x} \tag{2.30}$$

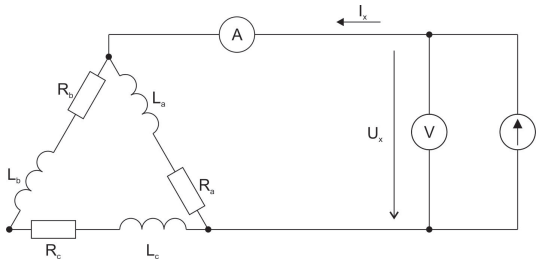


Figure 2.5: Stator resistance measurement scheme

From this measurement we get the resistance of the current path between two of the stator windings terminals R_x. Measuring the resistance between all of the windings ter-

minals validates the windings symmetry. As the stator windings are delta connected, the resulting stator resistance R_s is calculated as in 2.31. The resistance is calculated as an average value from the combinations shown in table 2.3.

$$R_s = \frac{3}{2} R_x \tag{2.31}$$

Terminals	$\mathbf{U_x(V)}$	$\mathbf{I_x(A)}$	$\mathbf{R_x(\Omega)}$	$\mathbf{R_{x20}(\Omega)}$
$A-B$	0.423	0.749	0.565	0.563
$A-C$	0.424	0.748	0.566	0.564
$B-C$	0.421	0.749	0.563	0.561
$A-B$	0.564	0.998	0.565	0.563
$A-C$	0.564	0.998	0.565	0.564
$B-C$	0.563	0.998	0.564	0.562

Table 2.3: Stator resistance measurement

The winding temperature is $21.10°C$. The result is:

- Stator Terminal Resistance R_x: 0.565 (Ω)
- Stator Terminal Resistance R_{x20}: 0.563 (Ω)
- Stator Phase Resistance R_s: 0.847 (Ω).

2.1.3.4 No-Load Test

The parameters obtained from no-load test and short-circuit test are measured and calculated at nominal voltage point. The parameters of the equivalent scheme correspond to phase values. A single phase apparent power is calculated as shown in equation 2.32.

$$S = U_f I_f \tag{2.32}$$

where U_f is phase voltage and I_f is phase current. As the stator winding is delta connected, the phase voltage and current are calculated as shown in 2.33 and 2.34.

$$U_f = U \tag{2.33}$$

$$I_f = \frac{I}{\sqrt{3}} \tag{2.34}$$

Where U is the delta voltage between two of the stator winding's terminals and I is the current flowing into the terminal. The three-phase apparent power is calculated as shown in equation 2.35.

$$S = 3U_f I_f = \sqrt{3} U I \tag{2.35}$$

There are a lot of properties that are derived from the no-load test [9][10]. For the induction motor model, we observe mainly the magnetizing reactance X_m and the mechanical losses P_m.

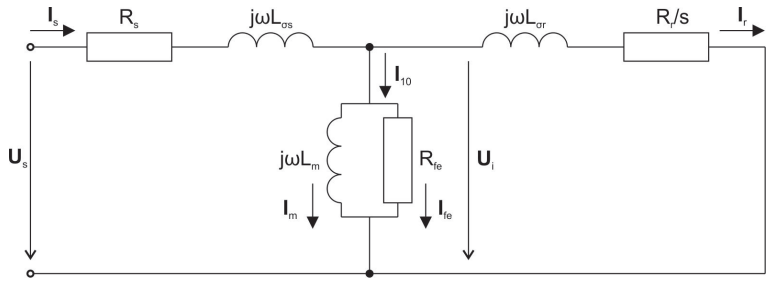


Figure 2.6: Induction motor steady-state equivalent circuit

While no load is connected to the shaft of the tested induction motor, the rotation speed of the rotor is near the synchronous speed. Note that the rotation speed can never be equal to the synchronous speed while working in motor mode due to the mechanical losses. The rotor resistance is inversely proportional to the slip s. The slip is defined as in 2.36[11].

$$s = \frac{\omega_{sl}}{\omega_s} = \frac{\omega_s - \omega_1}{\omega_s} \tag{2.36}$$

Where ω_{sl} is the slip angular speed, ω_s is the synchronous angular speed and ω_1 is the rotor angular speed. Assuming the rotor speed is near synchronous speed, we get equation 2.37.

$$\lim_{s \to 0} \frac{R_r}{s} = \infty. \tag{2.37}$$

Under this condition, the rotor circuit can be considered as open circuit. So the equivalent circuit of the induction motor steady-state shown in figure 2.6 is reduced as shown in figure 2.7.

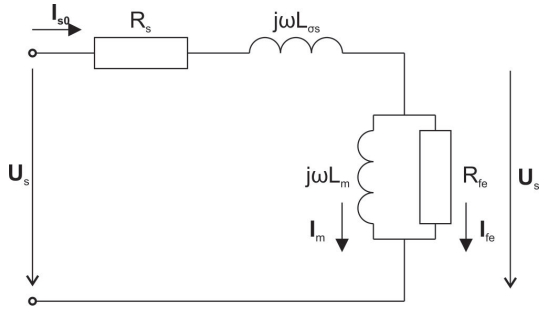


Figure 2.7: No-load reducer equivalent circuit

The scheme in figure 2.7 is described by equations 2.38 and 2.39.

$$\mathbf{U_s} = (R_s + j\omega_s L_{\sigma s})\mathbf{I_{S0}} + \mathbf{U_i} \tag{2.38}$$

$$\mathbf{I_{S0}} = \mathbf{I_m} + \mathbf{I_{Fe}} \tag{2.39}$$

Where $\mathbf{U_s}$ is the stator voltage phasor, $\mathbf{I_{s0}}$ is the stator current phasor, $\mathbf{U_i}$ is the induced voltage phasor, R_s is the stator resistance, L_m is the magnetizing inductance, $L_{\sigma s}$ is the stator leakage inductance and R_{Fe} is the resistor representing core and eddy current losses. These losses are not taken into account in the model of induction motor so the current $I_{Fe} = 0$. Under this condition, we can calculate the magnetizing inductance as shown in equation 2.40.

$$L_m = \frac{1}{2\pi f_{sn}} \frac{\sqrt{3}U_i}{I_{S0}} \tag{2.40}$$

Where U_i is the effective value of induced voltage, I_{S0} is the effective value of stator current and f_{sn} is the nominal frequency. The induced voltage is calculated from equation 2.38. The result of the no-load test is summarized in table 2.4.

f_{sn}	50	Hz
U_{0n}	400	V
I_{0n}	8.57	A
P_0	444.367	W
P_{j0}	62.20	W
P_m	106.68	W
P_{Fe}	275.49	W
$cos\varphi_{0n}$	0.075	$-$
L_M	249.304	mH

Table 2.4: No-load test result

2.1.3.5 Short-Circuit Test

The rotor of the induction motor is blocked during the test, so the test is also called *Blocked Rotor Test* ($\omega_1 = 0$). From equation 2.36 follows $s = 1$. As the impedance of the magnetizing branch is much higher than the impedance of the rotor, the magnetizing branch is neglected. Under this condition, the short-circuit equivalent scheme is simplified as shown in figure 2.8.

As the motor is short-circuit, nominal voltage cannot be used to supply the stator windings, otherwise, the resulting current could burn the insulation of the windings. So

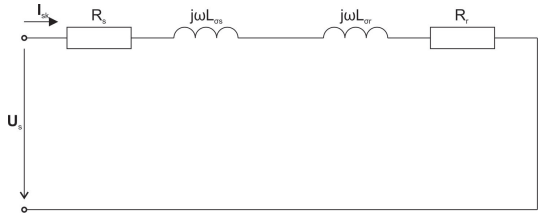


Figure 2.8: Short-circuit reducer equivalent circuit

the parameters are measured at lower voltage than nominal and extrapolated to the point of nominal voltage. The scheme in figure 2.8 is described by equation 2.41.

$$\mathbf{U_s} = ((R_s + R_r) + j\omega_s(L_{\sigma s} + L_{\sigma s}))\mathbf{I_{Sk}} \tag{2.41}$$

The rotor resistance is calculated as shown in equation 2.42.

$$R_r = \frac{P_k}{I_{kn}^2} - R_s = \frac{\sqrt{3}U_n I_{kn} cos\varphi_k}{I_{kn}^2} - R_s \tag{2.42}$$

The leakage inductance of stator and rotor cannot be measured separately. The leakage inductance measured in short-circuit test involves both the stator and the rotor leakage inductance values. To get the stator and rotor leakage inductance values $L_{\sigma s}$, $L_{\sigma r}$, the total leakage inductance is split into 55 : 45 ratio. The total leakage inductance is calculated in equation 2.43.

$$L_k = \frac{1}{2\pi f_{sn}} \frac{Q_k}{I_{kn}^2} = \frac{1}{2\pi f_{sn}} \frac{\sqrt{3}U_n I_{kn} sin\varphi_{kn}}{I_{kn}^2} \tag{2.43}$$

The relation between the total and stator leakage inductance is described in 2.44, relation between the total and rotor leakage inductance is in equation 2.45.

$$L_{\sigma s} = 0.55 L_k \tag{2.44}$$

$$L_{\sigma r} = 0.45 L_k \tag{2.45}$$

2.1.4 Model Validation

The validation process consists of steady-state and transient response validation. To validate the steady-state, stator currents measured in short-circuit test and no-load test must fit the model as well as the starting torque. In order to run the short-circuit test in the model, the velocity input of the induction motor subsystem must be disconnected. The results of the SC and NL tests are listed in table 2.6. The starting torque of the model is displayed in figure 2.9.

The dynamics of the model identified in section 2.1.3 are validated by measurement of transient response of speed and current on direct switch-on from 3x400V power supply. A

f_{sn}	50	Hz
U_n	400	V
I_{kn}	143.44	A
R_k	3.020	Ω
$cos\varphi_{kn}$	0.313	$-$
M_z	83.45	Nm
$L_{\sigma s}$	8.031	mH
$L_{\sigma r}$	6.425	mH
R_r	0.6648	Ω

Table 2.5: Short-circuit test result

Quantity	**Measured**	**Model**
$I_{f0}(A)$	4.95	4.95
$I_{fk}(A)$	82.82	83.83
$M_z(Nm)$	83.45	84.60

Table 2.6: Steady-state validation

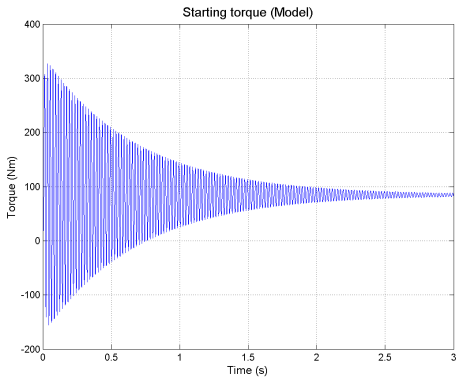
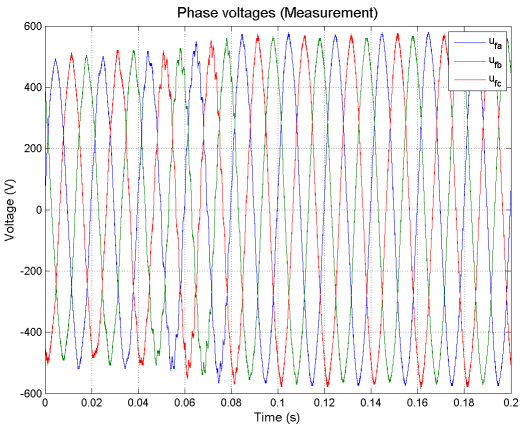
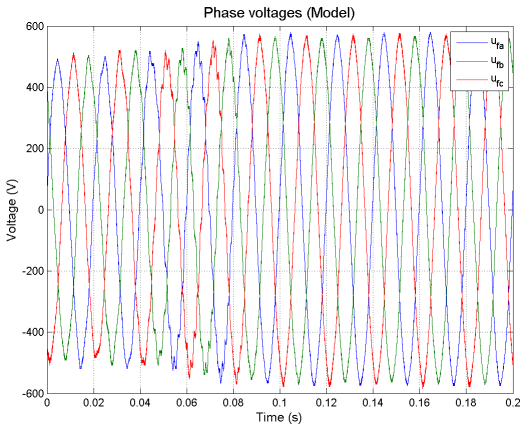


Figure 2.9: Model starting torque

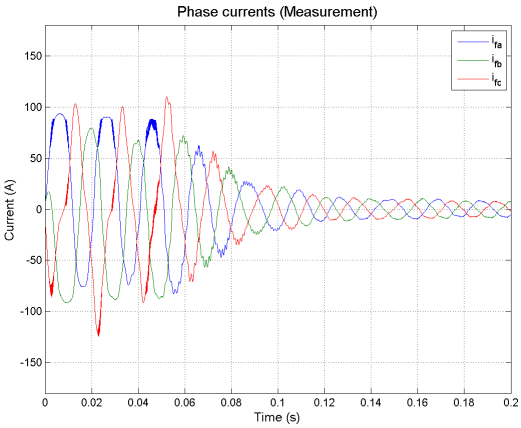
voltage drop occurs during the starting as the supply source does not have infinite short-circuit power. The same voltage is used to supply the model of the induction motor. The current and speed responses of the measured induction motor and the model of induction motor are displayed in figure 2.10.



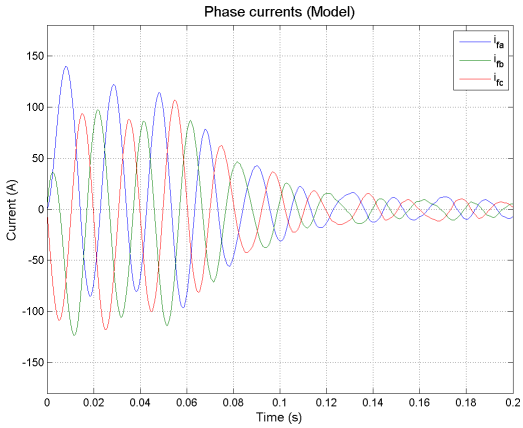
(a) Measured phase voltages



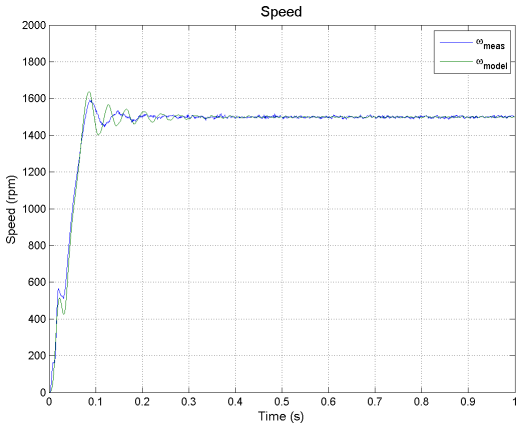
(b) Model phase voltages



(c) Measured phase currents



(d) Model phase currents



(e) Measured and model speed

Figure 2.10: Transient response validation

2.2 Power Inverter

To apply any of the advanced control methods of the induction motor, it is necessary to use a power inverter as a power supply. In these methods, every single winding is powered independently and precisely resulting in determined behaviour. To power a 3-phase induction motor with squirrel cage, 3-phase power inverter is used, as figure 2.11 shows.

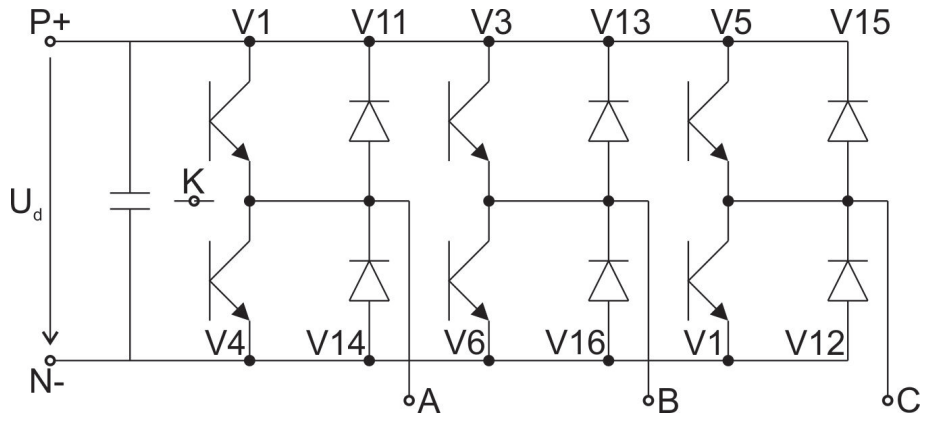


Figure 2.11: Power inverter

The power inverter in the scheme has a filter capacitor connected to the terminal of the DC bus. The DC power supply and the capacitor provide together power to the load connected to the AC terminals A, B, C by energy pulses. The space vector is used to describe the waveform of the voltage at AC terminals u_a, u_b, u_c [12].

2.2.1 Stator Voltage Space Vector

As windings of the induction motor stator are sinusoidally distributed along the rotor, voltage space vector is expressed as 2.46.

$$\overrightarrow{u}_s(t) = u_a(t)e^{j0} + u_b(t)e^{j\frac{2\pi}{3}} + u_c(t)e^{j\frac{4\pi}{3}} \tag{2.46}$$

The power inverter in figure 2.11 can be simplified by substituting the semi-conductor parts by switches switching between upper and lower bus, as shown in figure 2.12.

Where the q_a, q_b, q_c are control signals of these switches. Now, it is necessary to determine whether the load is wye or delta connected. The motor used in this system is delta connected. Focusing on the delta connected load, the voltages applied on the windings u_a, u_b, u_c is expressed as 2.47.

$$\begin{aligned} u_a &= u_A - u_B \\ u_b &= u_B - u_C \\ u_c &= u_C - u_A \end{aligned} \tag{2.47}$$

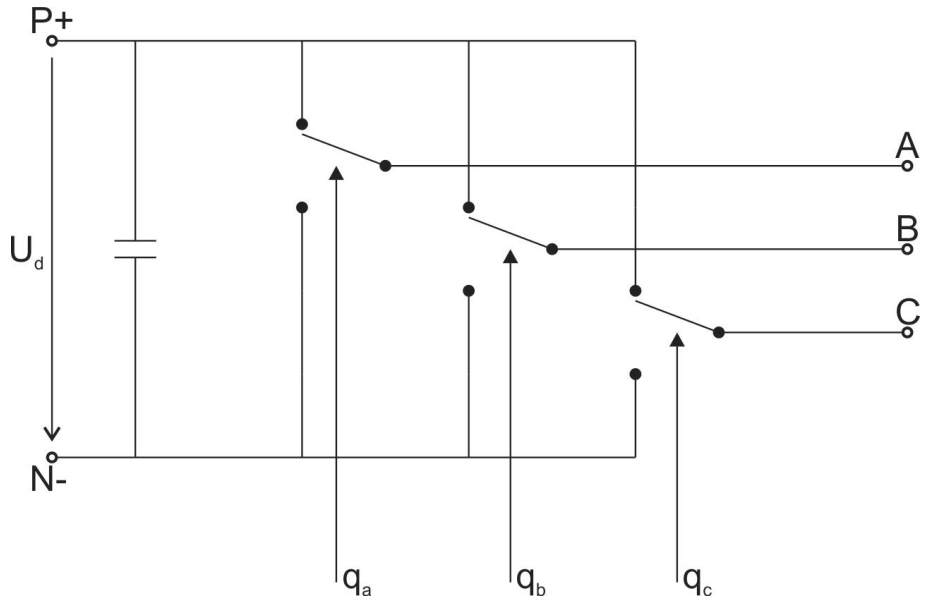


Figure 2.12: Simplified power inverter

Where u_A, u_B, u_C are the windings terminal voltages. The terminal voltage assumes $\{+U_d/2, -U_d/2\}$. Substituting from equation 2.47 into 2.46 results in

$$\overrightarrow{u}_s(t) = u_a e^{j0} + u_b e^{j\frac{2\pi}{3}} + u_c e^{j\frac{4\pi}{3}}. \tag{2.48}$$

This equation is written using logical signals of switches q_a, q_b, q_c as

$$\overrightarrow{u}_s(t) = \frac{U_d}{2}((q_a(t) - q_b(t))e^{j0} + (q_b(t) - q_c(t))e^{j\frac{2\pi}{3}} + (q_c(t) - q_a(t))e^{j\frac{4\pi}{3}}). \tag{2.49}$$

From this equation, it follows that there are 8 possible switching combinations which result in 7 different instantaneous voltage space vector positions. To define these space vector positions, the stator voltage space vector is defined as a function of logical signals of switches as described in equation 2.50.

$$\overrightarrow{u}_s(t) = f(q_c(t), q_b(t), q_a(t)) \tag{2.50}$$

All possible combinations of the stator voltage space vector positions are listed in 2.51 and displayed in figure 2.13.

$$\begin{aligned}
\overrightarrow{u}_s(000) &= \overrightarrow{u}_0 = 0 \\
\overrightarrow{u}_s(001) &= \overrightarrow{u}_1 = \sqrt{3}U_d e^{j\frac{\pi}{6}} \\
\overrightarrow{u}_s(010) &= \overrightarrow{u}_2 = \sqrt{3}U_d e^{j\frac{5\pi}{6}} \\
\overrightarrow{u}_s(011) &= \overrightarrow{u}_3 = \sqrt{3}U_d e^{j\frac{3\pi}{6}} \\
\overrightarrow{u}_s(100) &= \overrightarrow{u}_4 = \sqrt{3}U_d e^{j\frac{9\pi}{6}} \\
\overrightarrow{u}_s(101) &= \overrightarrow{u}_5 = \sqrt{3}U_d e^{j\frac{11\pi}{6}} \\
\overrightarrow{u}_s(110) &= \overrightarrow{u}_6 = \sqrt{3}U_d e^{j\frac{7\pi}{6}} \\
\overrightarrow{u}_s(111) &= \overrightarrow{u}_7 = 0
\end{aligned} \tag{2.51}$$

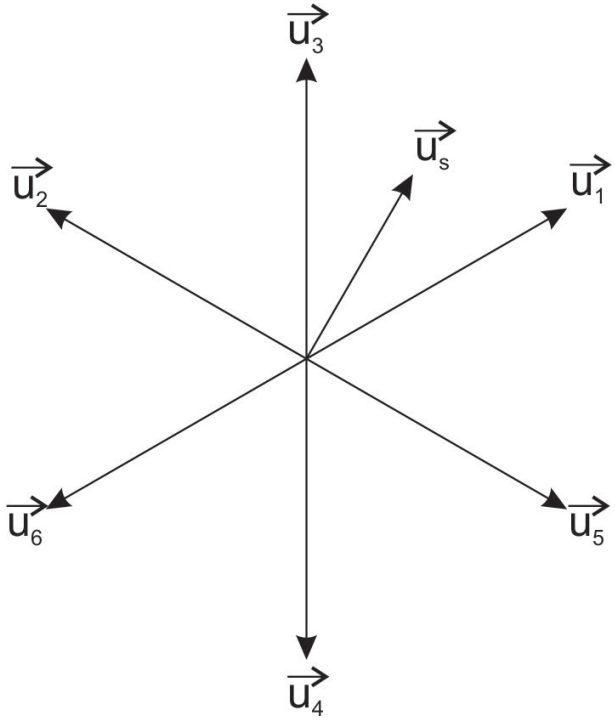


Figure 2.13: Stator voltage space vectors

2.2.2 Current flow

The input current of the power inverter i_{PI} is calculated as shown in equation 2.52.

$$i_{PI} = i_A q_a + i_B q_b + i_C q_c \tag{2.52}$$

Where i_A, i_B, i_C are the currents flowing through the power inverter terminals. The DC link input is filtered by a RLC filter. The filter circuit is displayed in figure 2.14. The circuit equations of the filter are shown in 2.53.

$$\begin{gathered} i_b = i_{RC} + i_{PI} \\ i_{RC} = i_{Cf} + i_{Rf} \\ u_B = u_{Lf} + u_{RC} \\ u_{Lf} = L\frac{di_L}{dt} \\ i_{Cf} = C\frac{du_{RC}}{dt} \\ u_{RC} = R_f i_{Rf} \end{gathered} \tag{2.53}$$

The input current of the filter is calculated using equations 2.52 and 2.53. This current is used as a load current of the battery pack.

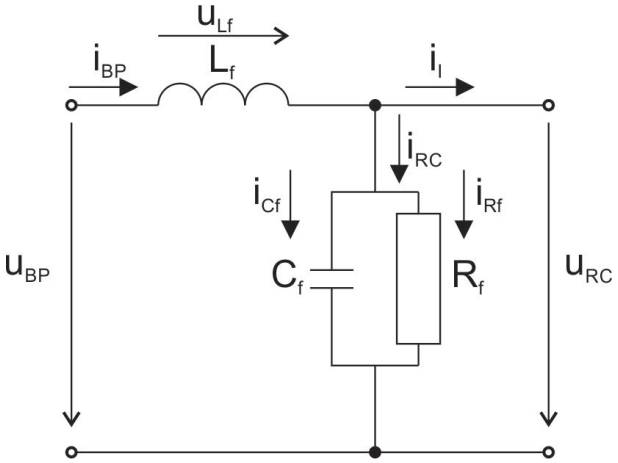


Figure 2.14: DC link input filter

2.3 Direct torque control

The direct torque control is a control strategy of power inverter powering AC motors which is used in dynamical demanding applications. This method, as well as the vector control method, provides independent control of motor torque and stator magnetic flux. Unlike vector control, DTC uses simple hysteresis (or bang-bang) regulators to regulate the torque and stator magnetic flux. On the other hand, DTC places higher demands on the computing performance of the controlling system than the vector control does. There are two different types of DTC named after their inventors, Manfred Depenbrock and Isao Takahashi. Both methods are based on direct regulation of electromagnetic torque in $\pm M$ band. The stator flux linkage space vector moves along a prescribed trajectory. Both methods use α, β transformation described in section 2.1.1.1. Instantaneous value of the electromagnetic torque is given by equation 2.54[13][14].

$$M_{em} = \frac{3}{2} p(\psi_\alpha i_\beta - \psi_\beta i_\alpha) \tag{2.54}$$

Where i_α, i_β are transformed stator currents i_a, i_b, i_c and ψ_α, ψ_β are calculated from two phase-to-phase voltages u_{ab}, u_{ac} and two phase currents i_a, i_c. The calculation and measurement of the voltage values and the current values must be fast enough to decide immediately which switch combination is necessary to apply to increase or decrease the electromagnetic torque. The motion of the stator flux linkage space vector along the prescribed trajectory is based on the equation of stator voltage as in 2.55.

$$\frac{d\overrightarrow{\psi}_{s,\alpha\beta}}{dt} = \overrightarrow{u}_{s,\alpha\beta} - R_s \overrightarrow{i}_{s,\alpha\beta} \tag{2.55}$$

By substitution of the derivative by difference we get equation 2.56.

$$\triangle \overrightarrow{\psi}_{s,\alpha\beta} = (\overrightarrow{u}_{s,\alpha\beta} - R_s \overrightarrow{i}_{s,\alpha\beta}) \triangle t \tag{2.56}$$

From equation 2.56, it follows that the direction of the stator flux linkage growth is identical to the direction of the applied stator voltage vector. The power inverter used in this system supports Takahashi DTC method, which is described in the following section.

2.3.1 Takahashi method

The Takahashi DTC method prescribes the motion of the end point of the stator flux linkage space vector in an area defined by two circles as shown in figure 2.15. The end point of the space vector moves along the circle trajectory. To achieve that, 6 basic stator voltage space vectors are used.
Example: End point of the stator flux linkage space vector is located in II. sector and its magnitude exceeds the upper limit. In order to continue in the positive direction (counterclockwise) and to decrease the magnitude of the flux linkage vector, the stator voltage u_6 is applied.

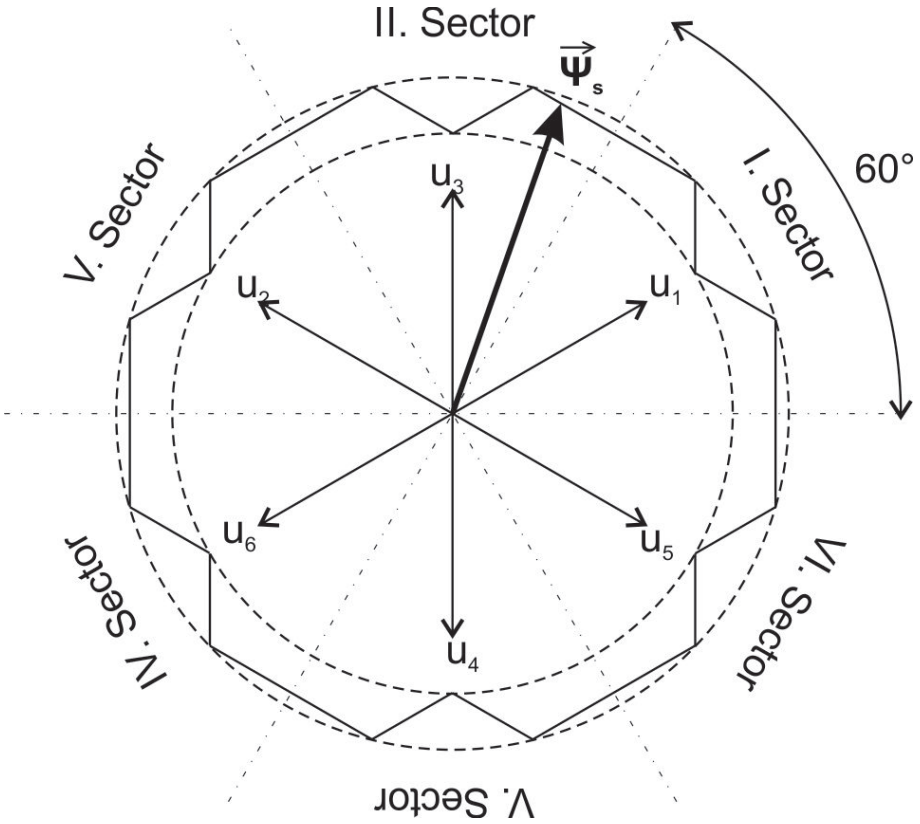


Figure 2.15: Takahashi stator flux linkage motion

There are 5 possibilities for the stator voltage vector to use in each sector as shown in figure 2.16. These possibilities are:

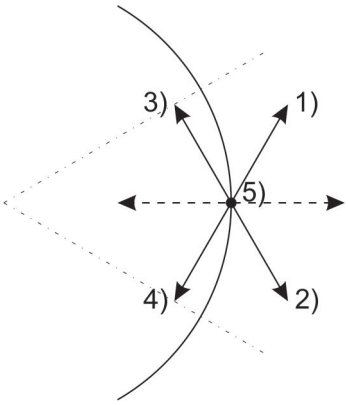


Figure 2.16: Takahashi motion possibilities

1. Out counter-clockwise direction resulting in electromagnetic torque increase and stator flux linkage magnitude increase.

2. Out clockwise direction resulting in electromagnetic torque rapid drop and stator flux linkage magnitude increase.

3. In counter-clockwise direction resulting in electromagnetic torque increase and stator flux linkage magnitude decrease.

4. In clockwise direction resulting in electromagnetic torque rapid drop and stator flux linkage magnitude decrease.

5. Zero voltage vector application resulting in slow electromagnetic torque decrease and constant stator flux linkage magnitude.;

The result of the applied stator voltage vector changes as the sector changes. It follows that we have to assign stator voltage vectors to the proper regulation demands for every single sector. The electromagnetic torque demand has higher priority than the stator flux linkage magnitude has. The regulation demands are the same for all sectors and are listed in table 2.7.

M_{em}		$\lvert\psi_s\rvert$	
↘	00	↘	0
↗	01	↗	1
↘↘	10		

Table 2.7: Regulation demands

Considering the torque demand bits as the most significant bits, and flux linkage magnitude demand as the least significant bit, we can define all combinations as shown in table 2.8.

M_{em}		$\lvert\psi_s\rvert$		sum (bin)	sum (dec)
↘	00	↘	0	000	0
↘	00	↗	1	001	1
↗	01	↘	0	010	2
↗	01	↗	1	011	3
↘↘	10	↘	0	100	4
↘↘	10	↗	1	101	5

Table 2.8: Regulation demands combined

Using table 2.8 it is possible to assign stator voltage vectors and their results as shown in table 2.9. The two zero vectors u_7 and u_0 are chosen in order to minimize switching losses.

The regulation scheme of the Takahaschi DTC method is shown in figure 2.17. The regulation of the electromagnetic torque and the stator flux linkage magnitude is described in the following section.

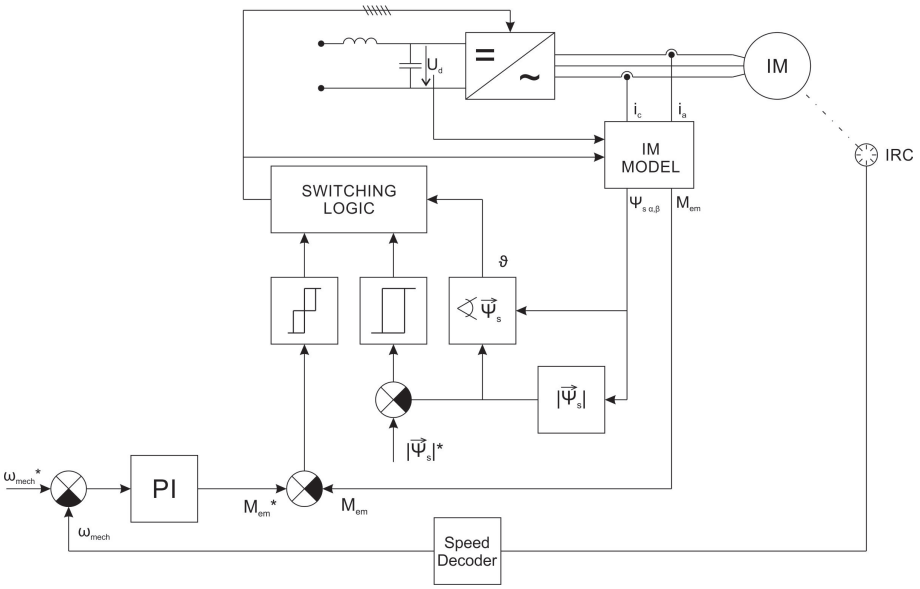


Figure 2.17: Takahashi DTC scheme

2.3.2 Regulators

The regulation scheme of the DTC is displayed in figure 2.17. From the scheme, it follows that the electromagnetic torque and magnitude of the stator flux linkage space vector are regulated separately using hysteresis regulators. The electromagnetic torque set point is set by the outer speed control loop. The error between the reference speed ω^*_{mech} and

Sector I.	
Demand	Voltage Vector
000 (0)	u_0 (000)
001 (1)	u_7 (111)
010 (2)	u_2 (010)
011 (3)	u_3 (011)
100 (4)	u_4 (100)
101 (5)	u_5 (101)

Sector II.	
Demand	Voltage Vector
000 (0)	u_7 (111)
001 (1)	u_0 (000)
010 (2)	u_6 (110)
011 (3)	u_2 (010)
100 (4)	u_5 (101)
101 (5)	u_1 (001)

Sector III.	
Demand	Voltage Vector
000 (0)	u_0 (000)
001 (1)	u_7 (111)
010 (2)	u_4 (100)
011 (3)	u_6 (110)
100 (4)	u_1 (001)
101 (5)	u_3 (011)

Sector IV.	
Demand	Voltage Vector
000 (0)	u_7 (111)
001 (1)	u_0 (000)
010 (2)	u_5 (101)
011 (3)	u_4 (100)
100 (4)	u_3 (011)
101 (5)	u_2 (010)

Sector V.	
Demand	Voltage Vector
000 (0)	u_0 (000)
001 (1)	u_7 (111)
010 (2)	u_1 (001)
011 (3)	u_5 (101)
100 (4)	u_2 (010)
101 (5)	u_6 (110)

Sector VI.	
Demand	Voltage Vector
000 (0)	u_7 (111)
001 (1)	u_0 (000)
010 (2)	u_3 (011)
011 (3)	u_1 (001)
100 (4)	u_6 (110)
101 (5)	u_5 (101)

Table 2.9: Switching logic tables

measured rotor speed ω_{mech} is amplified by a PI regulator to generate the torque reference M^*_{em}. These regulators are described in the following sections.

2.3.2.1 PI regulator

As already mentioned, the proportional-integral regulator is used in the outer speed regulation loop. The transfer function of the regulator is shown in equation 2.57.

$$G(s) = K_p(1 + \frac{1}{T_i}s) \tag{2.57}$$

The parameters of the regulator are the same for the model and the ABB power inverter used in the simulation system. The parameters are:

- $K_p = 7$

- $T_i = 2.5$

The speed regulated by the PI regulator using these values is displayed in figure 2.18.

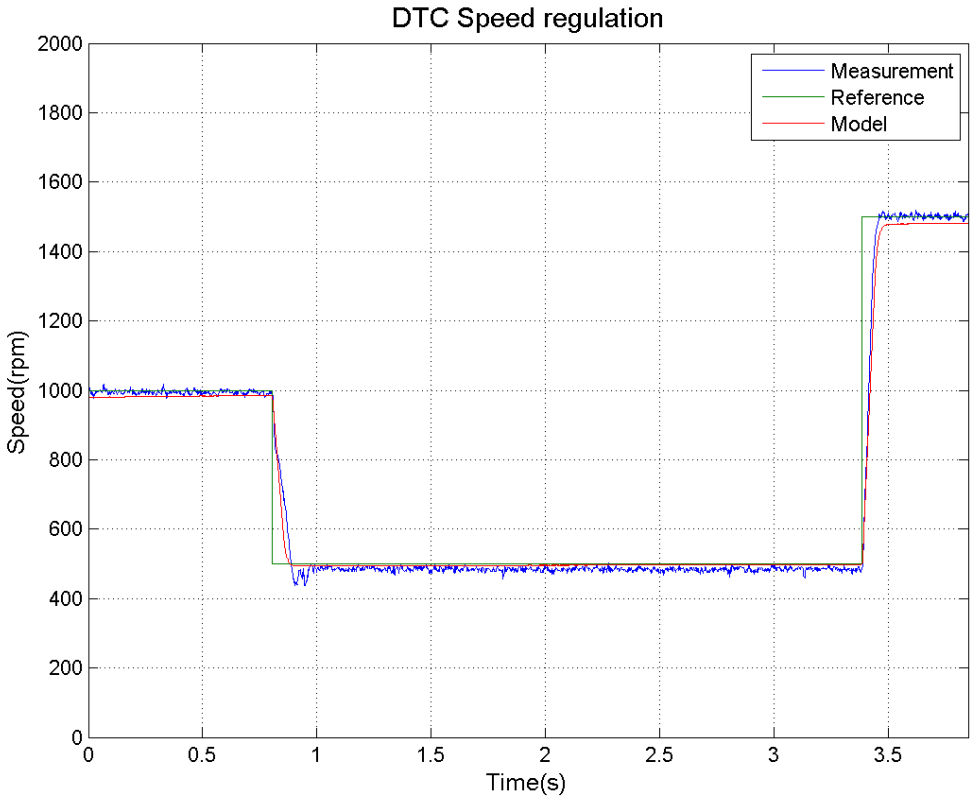


Figure 2.18: Speed regulation

2.3.2.2 Hysteresis regulators

The hysteresis regulator calculates the error value and watches whether the error value is in the band specified by the regulator. If the value crosses the upper or lower bound, the regulator sends the appropriate command to increase or decrease the error value. The band is usually set as ±5% of the rated value of the regulated quantity.

Such a simple regulator is used to regulate the magnitude of the stator magnetic flux linkage space vector. The regulator used to regulate the electromagnetic torque is more complex than the stator magnetic flux regulator. As there are 3 different commands to control the electromagnetic torque (see table 2.7), a three level hysteresis regulator is used. The three level hysteresis regulator is created by two basic hysteresis regulators, where the resulting command is the sum of commands of the basic hysteresis regulators. The hysteresis regulations are displayed in figure 2.19.

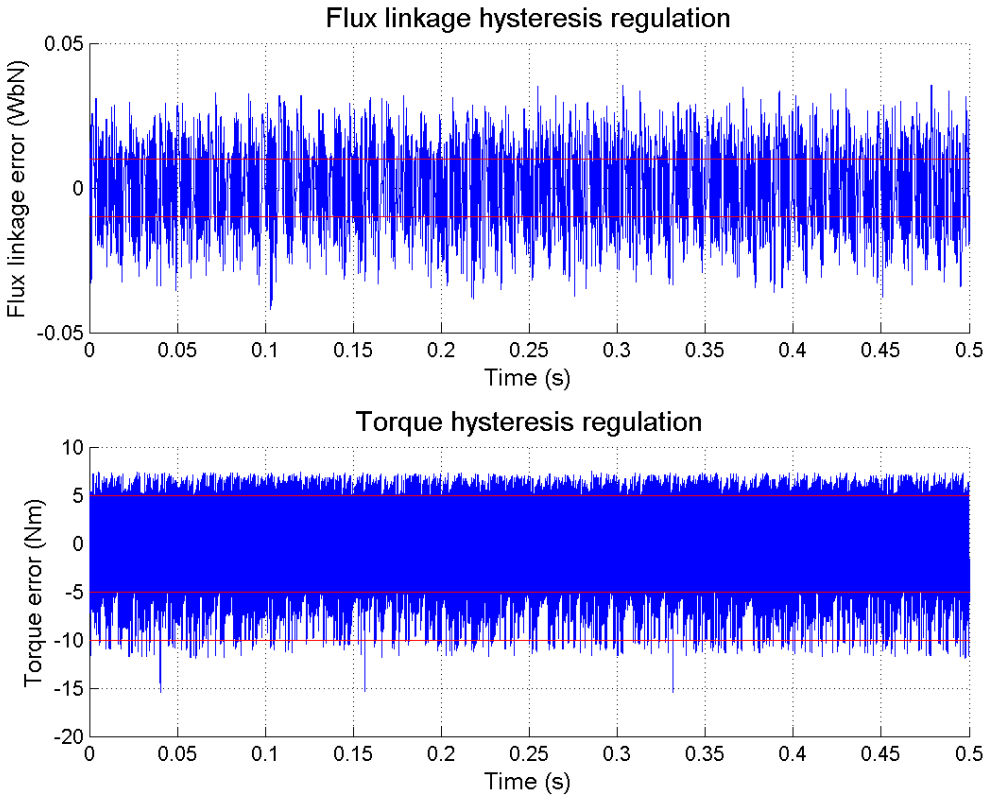


Figure 2.19: Hysteresis regulation

2.3.3 Validation

The DTC model validation consists of the validation of regulation, and comparison of the following: measured current, modelled current, measured stator flux linkage space vector trajectory and modelled stator flux linkage space vector trajectory. The flux and torque error hysteresis control are displayed in figure 2.19. The Takahashi DTC strategy prescribes circular trajectory of the end point of the stator flux space vector. The first trajectory of the end point of the reconstructed stator flux space vector from the measured stator currents and voltages of the real induction motor, as well as the second trajectory of the model, are displayed in figure 2.21. Measured and modelled stator phase currents are displayed in figure 2.20. This figure is a detail of the regulation displayed in figure 2.18.

2.4 Battery

A battery system is the most common way of power supply in automotive. A battery converts chemical energy directly into electrical energy. A key to choose a proper battery type for automotive industry is the energy density. Specific energy density of common

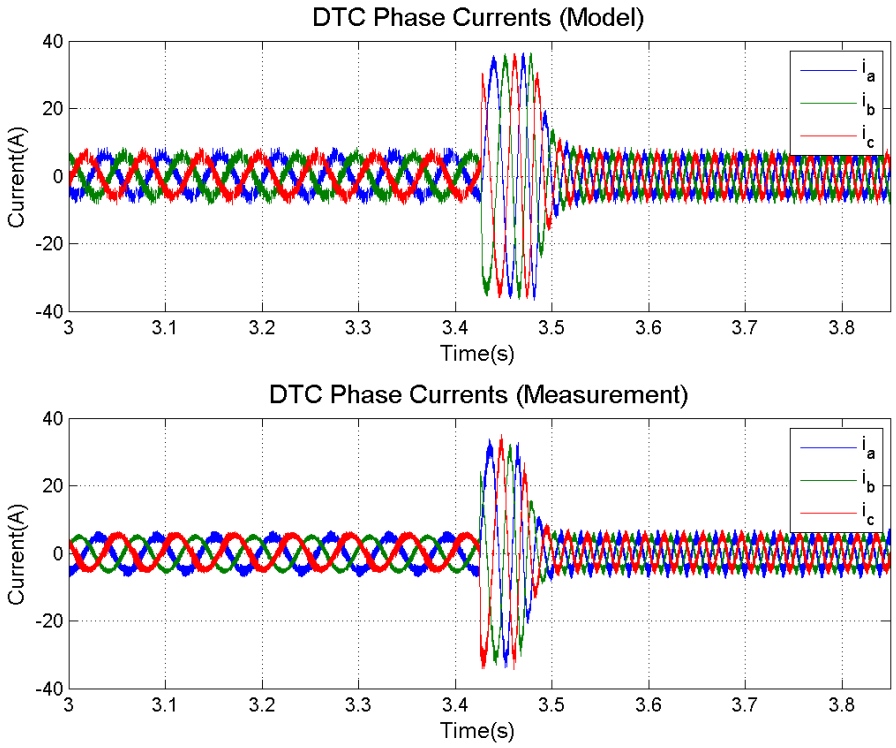


Figure 2.20: DTC phase currents

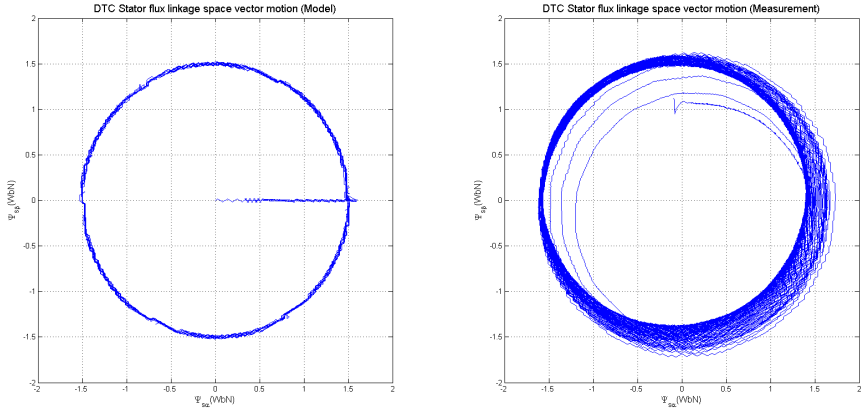


Figure 2.21: DTC stator flux linkage trajectory

battery types is in figure 2.22. This figure explains why the Lithium metal batteries are so popular in automotive industry.

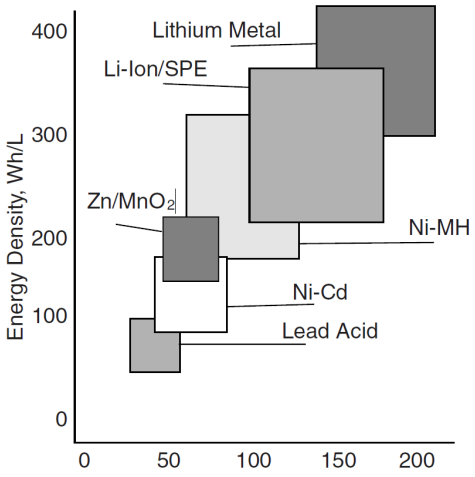


Figure 2.22: Specific energy density [1]

Specific values of Lithium Polymer battery cell are shown in table 2.10.

Property	**Value**	**Units**
Nominal voltage	$3.2 - 3.3$	$(V/cell)$
Charge up voltage	4.25	$(V/cell)$
Cut off voltage	2.7	$(V/cell)$
Charge current	$0.3 - 3., 0$	(CA)
Discharge current	$6(10Pulse)$	(CA)
Number of cycles	$2000(\%80DoD), 3000(\%70DoD)$	$(cycles)$
Self−discharge	3	$(\%/month)$
Operating temperature	$0.3 - 3.0$	$(^{\circ}C)$

Table 2.10: Specific properties of Li-ion battery cells

2.4.1 Battery model

There are various kinds of mathematical models of battery cell which describe various properties and behaviour. In this system, we would like to observe the power flow when it is discharging (or charging) the cell by pulse current. In this context, transient response is needed to be taken into account [15].

An ideal solution with satisfying accuracy would be an equivalent circuit with 2 RC networks.

Note: Battery cell properties change during discharging (respectively charging) according to state of charge (SoC), state of life (SoL), temperature, discharge current etc [16].

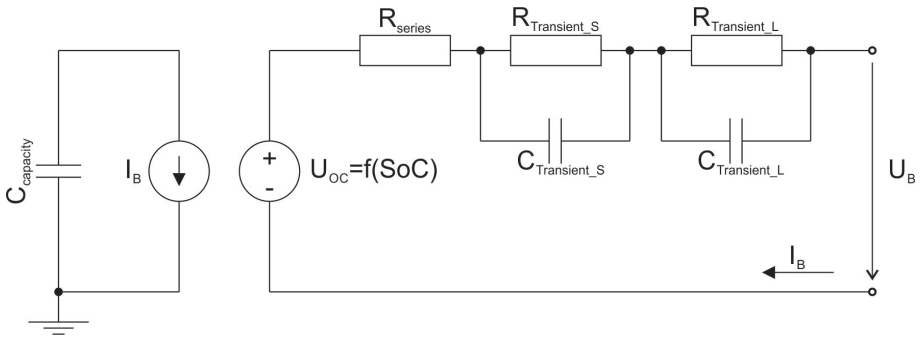


Figure 2.23: Battery cell equivalent circuit

According to [17], an equivalent circuit with 2 RC networks is, in automotive applications, a suitable trade-off between complexity and accuracy and simultaneously predicts runtime, steady-state, and transient response accurately. The state equation for the equivalent circuit is shown in equation 2.58.

$$\begin{bmatrix} \dot{SoC} \\ \dot{u_{RCS}} \\ \dot{u_{RCL}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{1}{R_{TS}C_{TS}} & 0 \\ 0 & 0 & -\frac{1}{R_{TL}C_{TL}} \end{bmatrix} \begin{bmatrix} SoC \\ u_{RCS} \\ u_{RCL} \end{bmatrix} + \begin{bmatrix} \frac{1}{Q_R} \\ \frac{1}{C_{TS}} \\ \frac{1}{C_{TL}} \end{bmatrix} I_B \tag{2.58}$$

Capacitor $C_{capacity}$ and current controlled current source on the left side of the equivalent circuit represent battery runtime. The voltage controlled voltage source, series resistor and RC networks respond to the current-voltage characteristic of the battery cell 2.23.

2.4.1.1 Usable capacity

Usable capacity represents the charge obtained from the battery cell when discharging from a fully charged state until the end of discharge voltage at cell terminal is reached. Usable capacity depends on the State of Life (SoL), discharge current, storage time (self-discharging), depth of discharge (DoD) and temperature. The influence of some of these phenomenons is not taken into account in this mathematical model as we focus on the power flow between the battery cell and the load. The state of charge is equal to 1 for fully charged state and is equal to 0 for fully discharged state. Self-discharging of the battery cell could be modelled by a resistor parallel to the capacitor $C_{Capacity}$. In our model, self-discharging is not taken into account.

2.4.1.2 Open-circuit voltage

Open circuit voltage drops while the battery cell is discharging. There is a non-linear relation between open circuit voltage and state of charge, which must be included in the model. This non-linear relation represents the function for output voltage of voltage source $U_{OC} = f(SoC)$ in Figure 2.23. We can measure this function as a steady-state open circuit voltage for different states of charge.

2.4.1.3 Transient response

Transient response of the battery cell consists of 3 parts.

- Instantaneous voltage drop across series resistor R_{Series}.
- Fast voltage drop across the RC_{TS}, which describes short time transient response
- Slow voltage drop across the RC_{TL}, which describes long time transient response

The inclusion of more RC networks makes model more accurate, however, increases complexity. As mentioned before, two RC networks are the best trade-off between accuracy and complexity. All these components are theoretical functions of SoC, SoL, discharge current and temperature [17].

2.4.1.4 Battery pack

In order to get appropriate voltage level in the DC link of the power inverter, it is necessary to configure the battery cells in series. The terminal voltage of the battery pack is calculated using equation 2.59.

$$U_{BP} = U_B n_s \tag{2.59}$$

Where U_{BP} is the terminal voltage of the battery pack, U_B is the terminal voltage of the battery cell and n_s is the number of battery cells in series. To increase the capacity of the battery pack, the series branches are configured in parallel. As the impedance of the branches is the same, the current splits between these branches as shown in equation 2.60.

$$I_B = \frac{I_B P}{n_p} \tag{2.60}$$

Where I_{BP} is the battery pack terminal current, I_B is the current flowing through one branch and n_p is the number of the parallel branches.
Note: The real battery cells do not have the same parameters, so the particular cells are not stressed equally. This can lead to a difference in the aging of cells. Moreover, it can cause failure of the system. To avoid these issues, the charging and discharging is managed by battery management system (BMS).
The battery cells in the mathematical model are considered as ideal so the battery management system is not modelled.

2.4.2 System identification

The values of the parameters of the equivalent circuit elements are obtained from the voltage transient response on the current discharge pulse. The transient response is mea-

sured for two different current pulses at different states of charge. The state of charge is established using equation 2.61.

$$SoC = \frac{1}{Q_c} \int i_{dis}(t)dt \tag{2.61}$$

Where Q_c is the capacity of the battery cell and $i_{dis}(t)$ is the discharge current. As the discharge current is constant, the SoC is calculated as shown in equation.

$$SoC = \frac{I_{dis}t}{Q_c} \tag{2.62}$$

The measurement is done in three steps:

1. The battery is discharged until the required SoC is reached.
2. The battery is disconnected till terminal voltage is stabilized.
3. The battery is loaded by short current pulse.
4. The voltage transient response is measured.

Considering the current pulse short, we can continue in the measurement repeating these steps until the end of discharge voltage is reached ($SoC = 0\%$). The parameters of the equivalent circuit are then calculated from the voltage transient response as shown in figure 2.24.

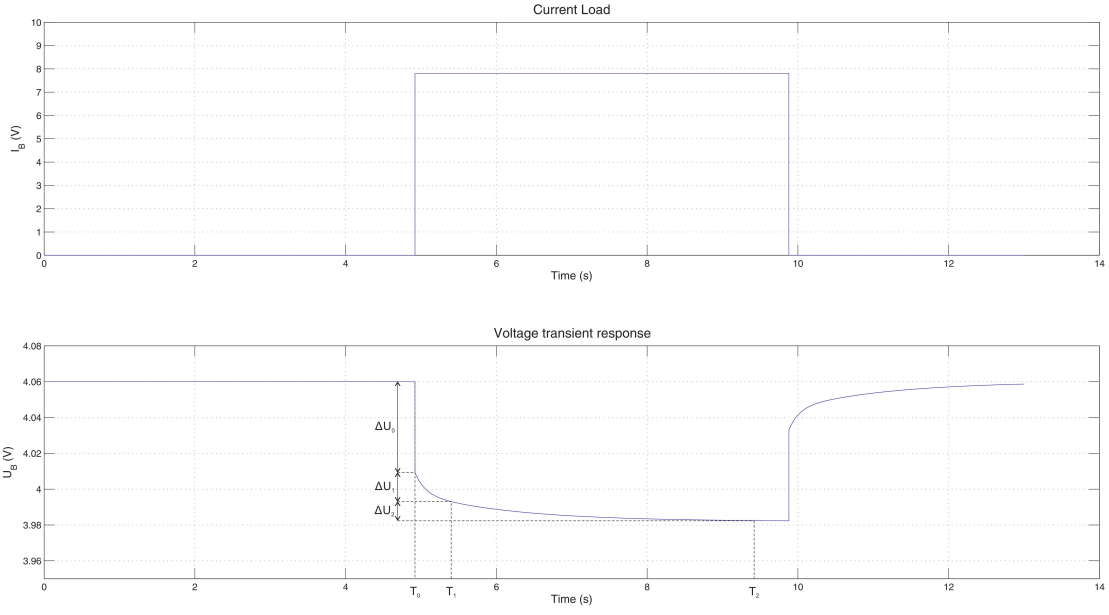


Figure 2.24: Parameters identification

From figure 2.24, the series resistance is calculated from instantaneous voltage drop ΔU_0 at time T_0, as shown in equation 2.63.

$$R_{Series} = \frac{\Delta U_0}{I_{dis}} \tag{2.63}$$

The resistances of the RC networks are then calculated as shown in equations 2.64 and 2.65.

$$R_{TS} = \frac{\Delta U_1}{I_{dis}} \tag{2.64}$$

$$R_{TL} = \frac{\Delta U_2}{I_{dis}} \tag{2.65}$$

The RC network capacities are calculated from time constants ($\tau = RC$). Considering the transient finished after time $T = 3\tau$, the capacity values are calculated as 2.66 and 2.67.

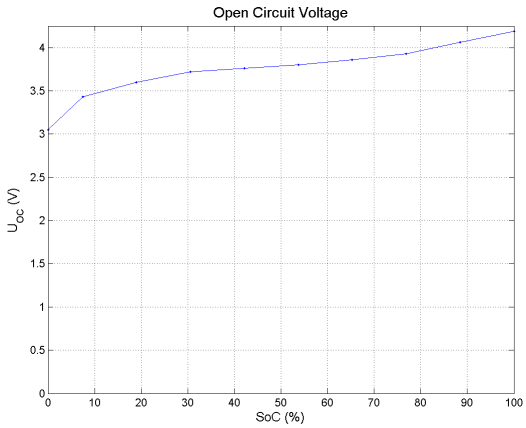
$$C_{TS} = \frac{T_1 - T_0}{3R_{TS}} \tag{2.66}$$

$$C_{TL} = \frac{T_2 - T_0}{3R_{TL}} \tag{2.67}$$

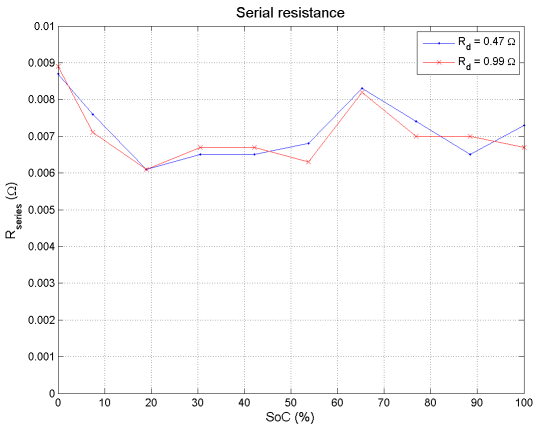
This system identification method is applied on Li-ion battery cell KOKAM SLPB11043140H4. The parameters are calculated for 9 different states of charge and 2 different current pulses. The result of the measurement is displayed in figure 2.25. These parameters are used in the mathematical model of the battery pack as look-up tables.

2.4.3 Validation

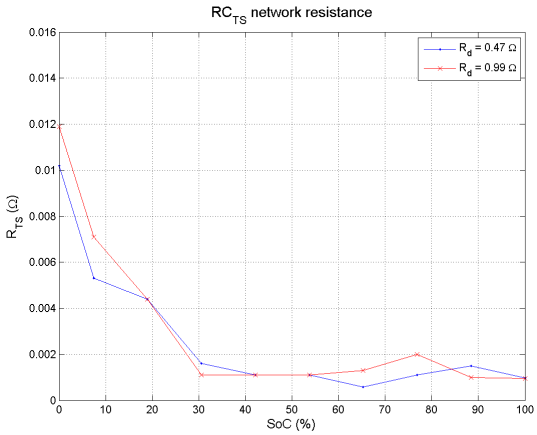
The model of the battery cell described in the previous section is validated by comparing the voltage transient response to the current load pulse with transient response of the model. The voltage transient response at a current pulse $I_{dis} = 7.4A$,and at a state of charge $SoC = 0.4$ is displayed in figure 2.26.



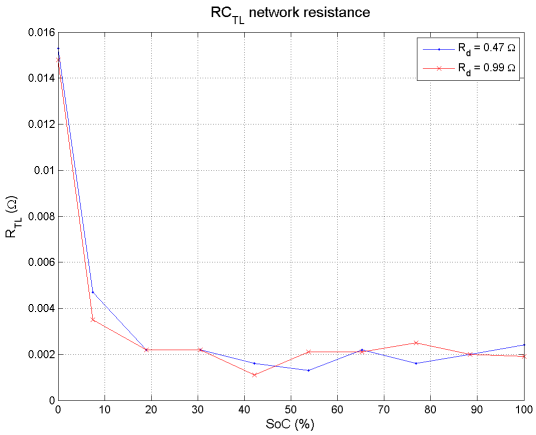
(a) Open circuit voltage



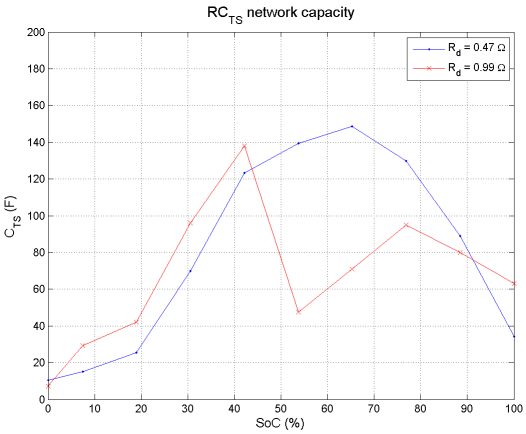
(b) Series resistance



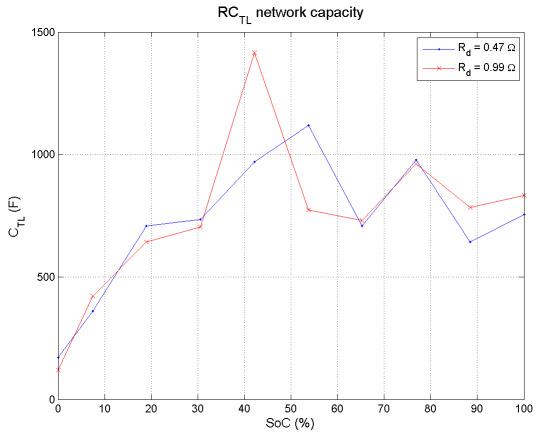
(c) Resistance of RC_{TS} network



(d) Resistance of RC_{TL} network



(e) Capacitance of RC_{TS} network



(f) Capacitance of RC_{TL} network

Figure 2.25: Measured parameters of the battery equivalent circuit

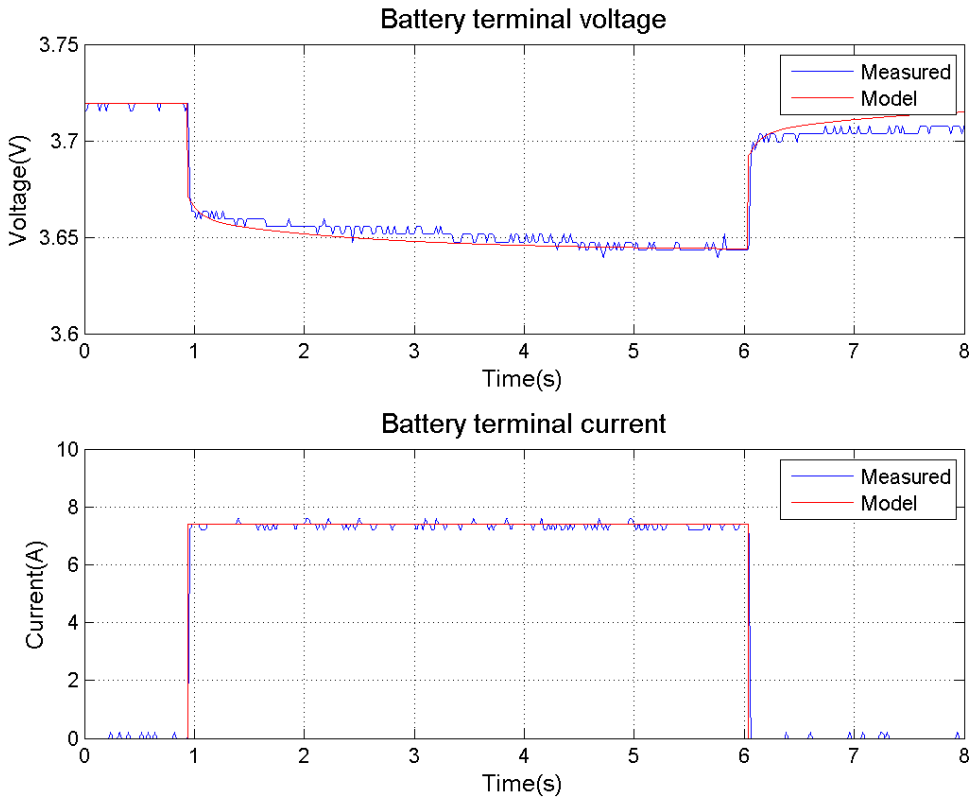


Figure 2.26: Battery model validation

2.5 Efficiency map

The dynamic model of the powertrain described in chapter 2 so far is useful to observe certain operation states and transition between these states. The accuracy of the dynamic model goes hand in hand with its complexity, so it cannot be used in mathematical iterative methods. These methods are used in optimizing. To minimize the calculation time of an iteration, the model of the powertrain must be simplified.

An efficiency map is a model describing efficiency of energy conversion in all operation states of the powertrain. The efficiency map can be used in optimizing iterative methods, where energy consumption is observed. It fulfils the requirement of low calculation time, however, it only describes efficiency of energy conversion in steady-state. Energy conversion process in powertrain is displayed in figure 2.27.

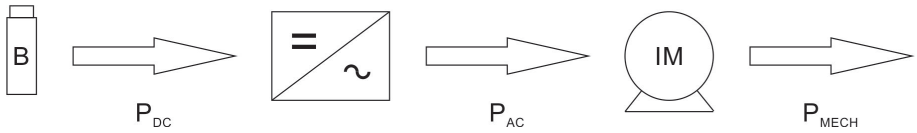


Figure 2.27: Powertrain energy conversion

An operating state of the powertrain is defined by the actual values of the torque on the shaft and the rotor angular speed of the induction motor. The efficiency map comes from measurement of all operating states where the efficiency of energy conversion is calculated. There are 3 energy converters in the powertrain. However, only the power inverter and the induction motor are modelled by the efficiency map.

2.5.1 Power inverter

The model of power inverter described in the previous section uses ideal loss-free switches. However, there are 3 basic kinds of losses of switches in the real power inverter:

- Losses in the switched on state
- Losses in the switched off state
- Switching losses

These losses are included in the efficiency map of the power inverter. The efficiency is calculated as shown in equation 2.68.

$$\eta_{PI} = \frac{P_{AC}}{P_{DC}} \tag{2.68}$$

The measured efficiency map of the power inverter used in the system is shown in figure 2.28.

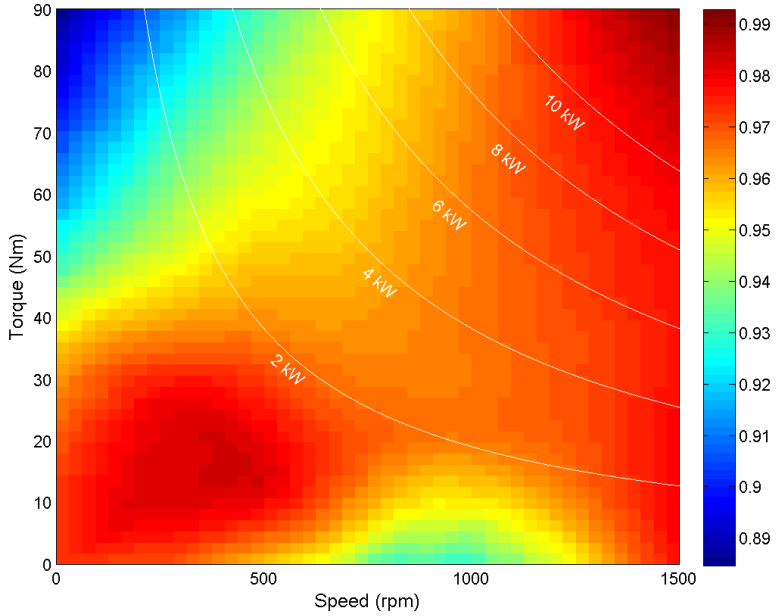


Figure 2.28: Power inverter efficiency map

2.5.2 Induction motor

The efficiency of induction motor energy conversion is calculated as shown in equation 2.69 and the efficiency map is displayed in figure 2.29.

$$\eta_{IM} = \frac{P_{MECH}}{P_{AC}} \tag{2.69}$$

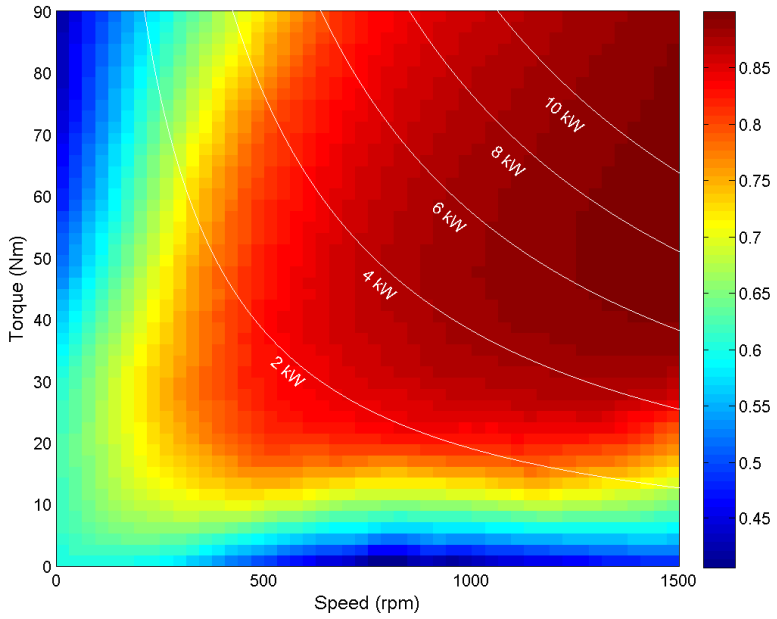


Figure 2.29: Induction motor efficiency map

2.5.3 Powertrain

The efficiency of the powertrain is calculated as shown in equation 2.70.

$$\eta_{PT} = \frac{P_{MECH}}{P_{DC}} = \eta_{PI}\eta_{IM} \tag{2.70}$$

The resulting efficiency map of the powertrain is shown in figure 2.30.
The efficiency map of the EV powertrain model is displayed in figure 2.31.

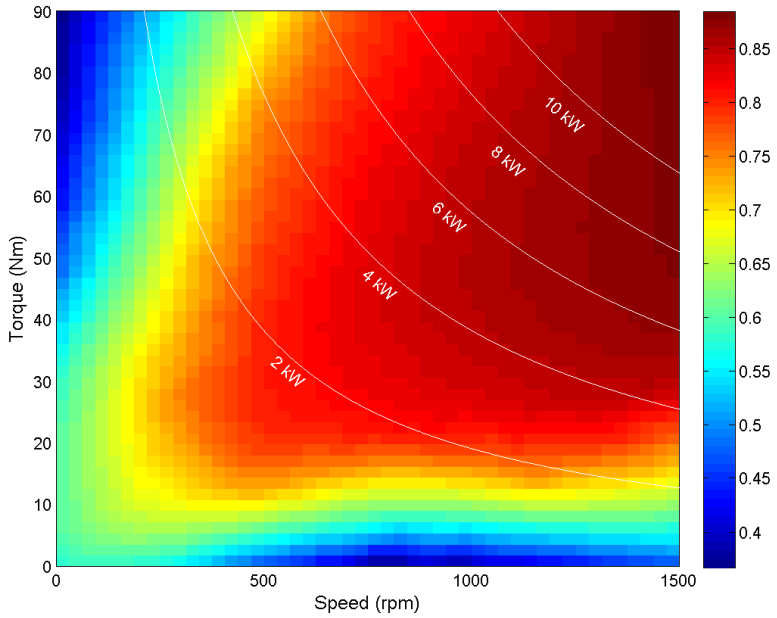


Figure 2.30: Powertrain efficiency map

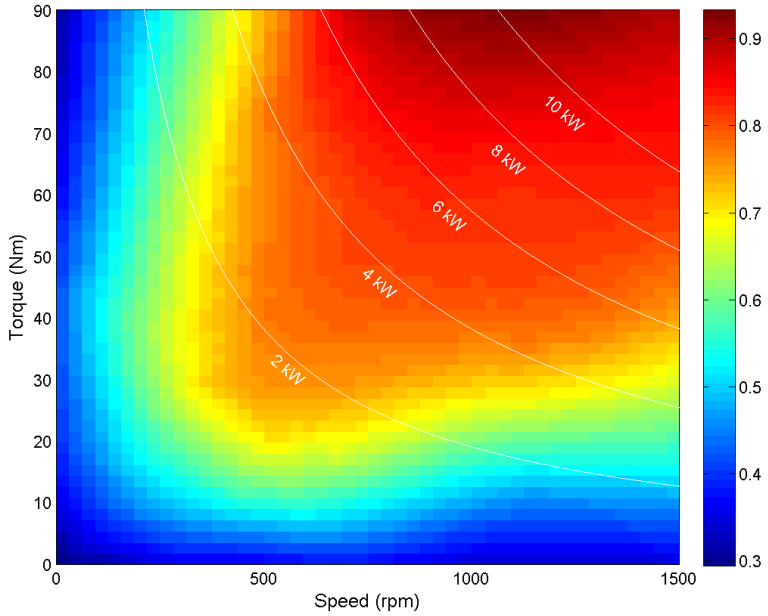


Figure 2.31: Powertrain model efficiency map

Chapter 3

Simulation system

In this chapter, the development of the simulation system is discussed. Firstly, the simulation system is introduced, then the installed hardware is described in Hardware Description section. This is followed by Communication protocols section describing communication establishment between all actuators. After the communication part, the developed control system and safety features are described in Control system and Safety Features sections.

3.1 Introduction

The simulation system simulates EV drive along a track with a known altitude according to a predefined speed of the vehicle and in respect to the drive resistance forces. The power flow and energy consumption are measured during the simulation. The measurement is used as a validation of the mathematical models of the EV.

In this chapter, the development of the simulation system is discussed. First, it is necessary to define the inputs of the simulation system, which are the EV parameters, speed profile and profile of the angle of inclination. The speed profile and the profile of the angle of inclination mean that the speed and the angle of inclination are functions of distance, as shown in figure 3.1.

The angle of inclination is calculated from the known altitude, and it is necessary to know the angle of inclination value in order to calculate the drive resistance forces. The calculation is described in section 3.4.1. The EV drive is simulated on a test bench controlled by dSpace DS1103. The development of the test bench control system is divided into 4 parts:

- Hardware installation
- Communication establishment
- Control system development
- Safety features implementation

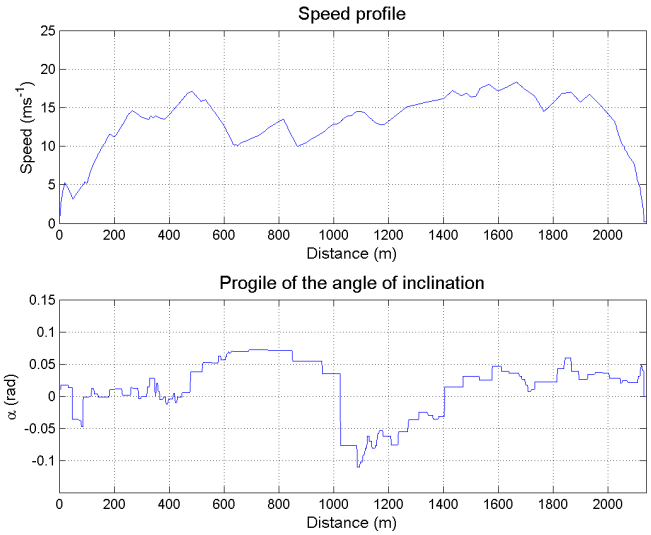


Figure 3.1: Speed and angle of inclination profiles

The scheme of the simulation system is displayed in figure 3.2.

3.2 Hardware description

The test bench at VTP Roztoky consists of an induction motor and a dynamometer. These two components are powered by power electronics. Only the power inverter supplying the induction motor is discussed in this thesis. The control system of the dynamometer is considered as a black box communicating by RS232 communication protocol. The hardware used in this system is listed in table 3.1. The induction motor and the power inverter simulate an EV powertrain. The DC link of the power inverter is powered by 6 pulse rectifier. The battery pack model parameters, measured in section 2.4, are necessary to run a battery simulator, which should replace the rectifier. The dynamometer simulates the drive resistance forces. These are calculated on-line in the control system and applied as a torque demand for the dynamometer. The control system is implemented in the dSpace DS1103.

3.2.1 Induction motor

The motor used in this system is a standard three-phase squirrel-cage induction motor with delta connected stator windings. The induction motor does not have enough power to drive the EV. This induction motor is used only to test the simulation system. The power of the induction motor has to be increased in a virtual way by decreasing the load torque reference for dynamometer and the speed reference for the ABB drive by reduction constant so as to meet the power requirement of the EV powertrain. The induction motor

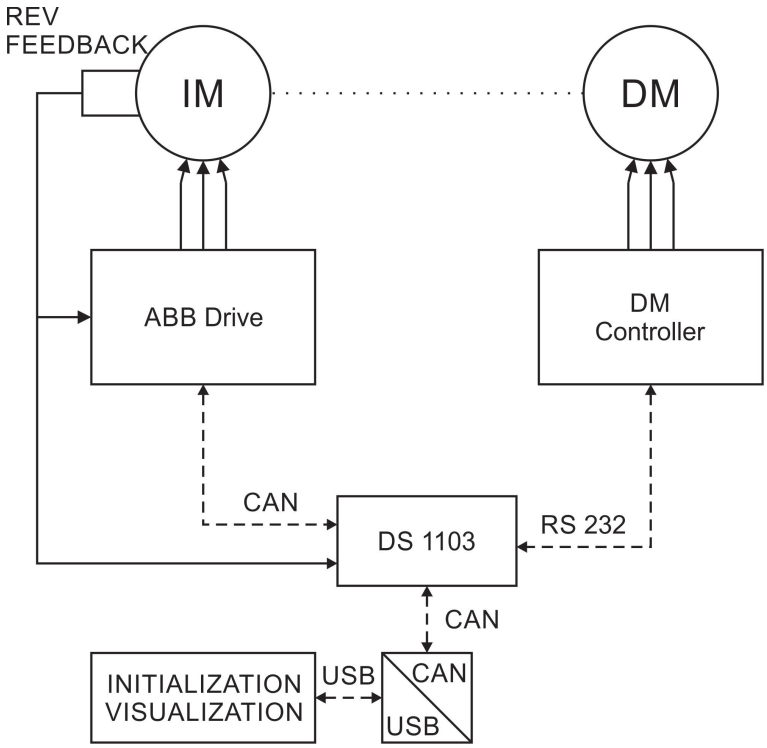


Figure 3.2: Test bench scheme

is equipped by a magnetic incremental encoder MSK320 [18]. The encoder specification is listed in table 3.2.

3.2.2 Power inverter

The power inverter ABB ACS880-01 uses DTC. This control method is not commonly used in automotive, however the power inverter is equipped by the FCAN-01 CANopen adapter module [4] and the FEN-31 HTL Encoder Interface [19]. The FCAN-01 CANopen adapter module provides control of the power inverter by CAN communication protocol and the FEN-31 HTL Encoder Interface decodes the signal from the incremental encoder measuring the induction motor speed, so it meets the requirements of our control system. The power inverter is displayed in figure 3.3.

FEN-31 HTL Encoder Interface FEN-31 is a module decoding the digital signal from the MSK320 incremental encoder. The module supports various types of output circuits of the encoder. To get valid speed value, the module has to be set according to the encoder type. A shortened module setting is listed in table 3.3.

FCAN-01 CANopen adapter FCAN-01 CANopen adapter is a module enabling communication with the power inverter by CANopen communication protocol. The communi-

Name	Property	Value	Units
Induction motor (F160M04 08L)	Power	11	kW
	Voltage	380	$V3\sim$
	Current	22	$A3\sim$
	Speed	1445	rpm
	Frequency	50	Hz
Dynamometer (ASD P085-4/6541)	Power	85	kW
	Voltage	400	$V3\sim$
	Current	147	$A3\sim$
	Speed	0-6000	rpm
	Inertia	0.445	kgm^2
Power Inverter (ABB ACS880-01)	Apparent power	26	kVA
	Input voltage	400	$V3\sim$
	Input current	38	$A3\sim$
	Input frequency	50/60	Hz
	Output voltage	0-Input voltage	$V3\sim$
	Output current	38	$A3\sim$
	Output frequency	0-500	Hz

Table 3.1: Installed hardware table

Feature	Technical data
Operating voltage	24 V DC
Output circuit	push-pull
Output signal	not inverted
Resolution	0.1 mm

Table 3.2: MSK320 label

Property	Value
Pulses per revolution	288
Pulse encoder type	Quadrature
Speed calculation mode	A&B all

Table 3.3: FEN-31 HTL encoder Interface settings

cation is described in section 3.3. The setting of the adapter is listed in table 3.4.

3.2.3 dSpace

The dSpace DS1103 is a controller board used for RCP (Rapid Control Prototyping) and HIL (Hardware-in-the-loop) software. The control model is developed in the Matlab/Simulink environment. Building the model in Simulink generates C code, which is

Figure 3.3: ABB ACS880-01 [2]

Property	Value
FBA Type	CANopem
Node ID	3
Bit rate	250 kb/s
Profile	CiA 402

Table 3.4: FCAN-01 CANopen adapter settings

then compiled and uploaded into the dSpace.

3.3 Communication protocols

Communication between all actuators is an essential part of every control system. There are three communication channels in the simulation system, as shown in figure 3.2. Two of them use CAN communication protocol. This is established between dSpace DS 1103 and power inverter ABB ACS880-01 as well as between DS 1103 and user PC. Communication between DS 1103 and ASD P085-4/6541 is done by RS 232. Both communication protocols are discussed in the following paragraphs.

Figure 3.4: dSpace DS1103 [3]

3.3.1 Controller Area Network

The Controller Area Network (CAN) is a serial multi-master bus standard typically used in automotive. CAN communication protocol can be divided into two basic layers. The lower layers include the data link and physical layer. The upper layers are referred to as higher-layer protocols.

Physical layer Physical layer is the physical connection between all nodes which is established by two wire bus terminated at each end by 120Ω resistor as figure 3.5 shows.

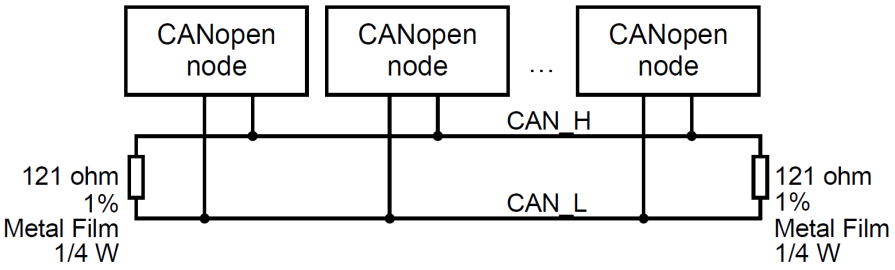


Figure 3.5: CAN physical connection [4]

CAN data link layer CAN data link layer in this application is based on the so called Classic CAN protocol internationally standardized in ISO 11898-1. CAN communication is based on broadcast so each node is allowed to transmit at any time. When two or more nodes start to transmit at the same time a communication collision can occur. To avoid this, an arbitration method is used based on the value of frame identifier (ID). A Frame with lower ID has higher priority so the frame with the lowest ID wins the arbitration and gets the bus access. A node that loses arbitration re-queues its frame for later transmission.

Data frame structure The data frame of Classic CAN starts with the Start of Frame (SOF) 1-bit field with a dominant bit level. The second is an Arbitration Field consisting of 11-bits Communication Object ID (COB ID) and 1-bit Remote Transmission Request (RTR). This is followed by Control Field with Data Field length information. Arbitration Field together with Control Field are called the Header. The next field after the header is the already mentioned 8-bytes Data Field followed by 16-bits Cyclic Redundancy Checksum (CRC) and 2-bits Acknowledge Field (ACK) indicating correct reception of the frame. The whole frame ends by 7-bits End of Frame. This is described in figure 3.6.

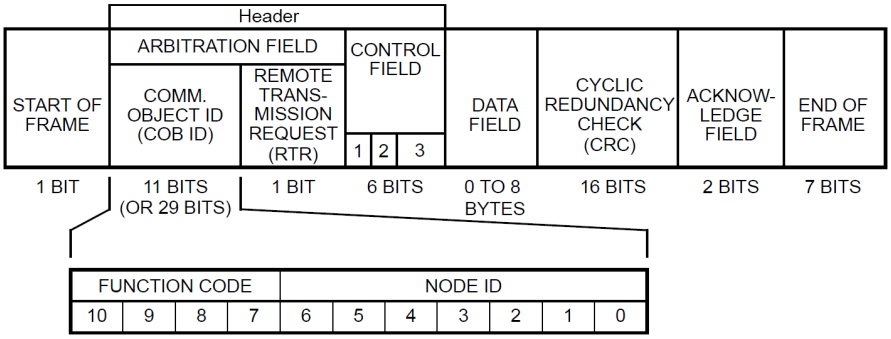


Figure 3.6: Data frame structure [4]

Application layer Application layer of CAN communication protocol in this application is defined by standardized CiA 402 CANOpen. This higher-layer protocol is described in greater detail in the following section [20].

3.3.1.1 CANOpen

CANOpen is a CAN-based higher-layer protocol and device profile which simplifies the development of control systems by standardised communication objects and device structure. CANOpen supports direct access to device parameters and transmission of time-critical process data. Every communication object is associated with one or more identifiers called Communication Object Identifier (COB ID) which defines implicitly its priority on bus as mentioned in the previous section. The COB ID consists of Function Code and Node ID and is defined as $((FunctionCode << 7) + NodeID)$. CANOpen defines all communication objects and information of their properties and functions. These objects are stored in Object Dictionary (OD) as shown in 3.5 which is saved in the device.

The OD is used as an interface between application and device. The COB can be reached by Service Data Object (SDO) and 16-bits index. If array type objects are to be reached, the index must be extended by a subindex.

Network management (NMT) The NMT message is mapped to a single CAN frame with COB ID 0 and 2-bytes data field consisting of NMT Command in the first byte and

Index (hex)	Object
0000	not used
$0001 - 001F$	Static Data Types
$0020 - 003F$	Complex Data Types
$0040 - 005F$	Manufacturer Specific Complex Data Types
$0060 - 007F$	Device Profile Specific Static Data Types
$0080 - 009F$	Device Profile Specific Complex Data Types
$00A0 - 0FFF$	reserved
$1000 - 1FFF$	Communication Profile Area
$2000 - 5FFF$	Manufacturer Specific Profile Area
$6000 - 9FFF$	Standardised Device Profile Area
$A000 - FFFF$	reserved

Table 3.5: Object Dictionary

Node ID in the second one. The NMT is sent by master node, thus forcing other nodes on the bus to go to equivalent NMT states. The NMT state machine is defined by CANOpen Communication Profile. The boot-up diagram of the state machine is shown in figure 3.7.

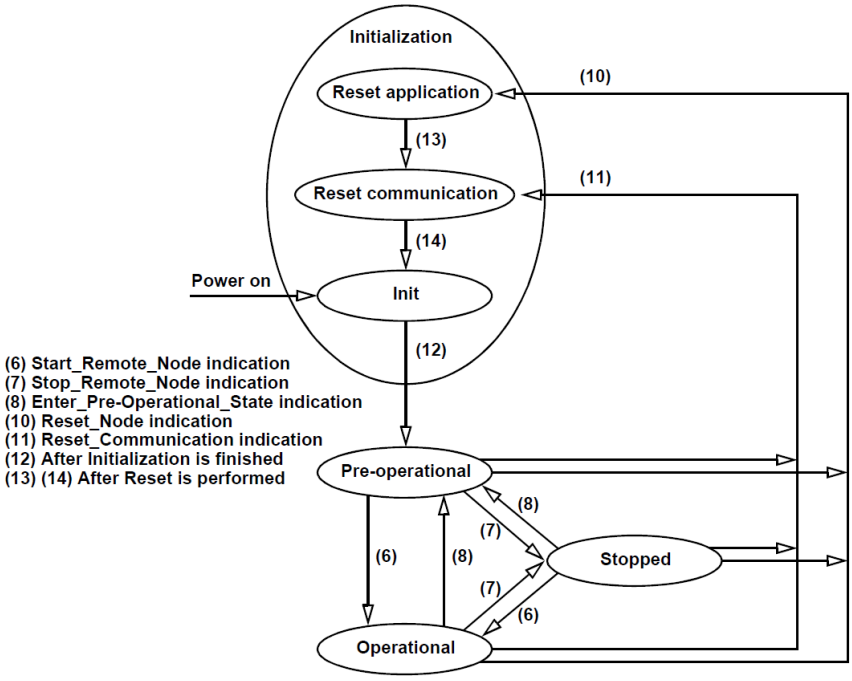


Figure 3.7: Boot-up diagran[4]

The NMT commands are listed in table 3.6.

Command (dec)	Name
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre-Operational State
129	Reset Node
130	Reset Communication

Table 3.6: NMT Commands

Service Data Object (SDO) The SDO is used to modify entries of the object dictionary by the following services:

- SDO Upload
- SDO Download
- Abort SDO Transfer

The SDO communication Client to Server is COB ID $600h + NodeID$, Server to Client is $580h + NodeID$. This type of communication object is disabled during Operational state.

Process Data Object (PDO) The PDO is enabled only in Operational state. It is used for time−critical process data exchange due to its high priority. Transmission of PDO can be triggered by request, internal timer or by received SYNC message. The PDO properties and application objects mapping are described in the OD. Mapping of PDOs can be configured by SDO in Pre-Operational state [21].

3.3.1.2 CAN messages

The CAN messages are defined using CANdb++ Editor as shown in figure 3.8. The objects defined in the CAN database can be easily imported into CAN communication blocks in Matlab/Simulink. The CAN messages used in the control system are listed in table 3.7.

3.3.2 RS232

RS232 is an asynchronous serial bus. The RS232 standard defines how to transmit certain bit sequence, however, it does not define higher communication layers. The data length is not limited. The data bits are transmitted in order from the LSB to the MSB. As the communication is asynchronous, a synchronization of the slave must be done before data transmission begins. It is done by synchronization message sent by master.

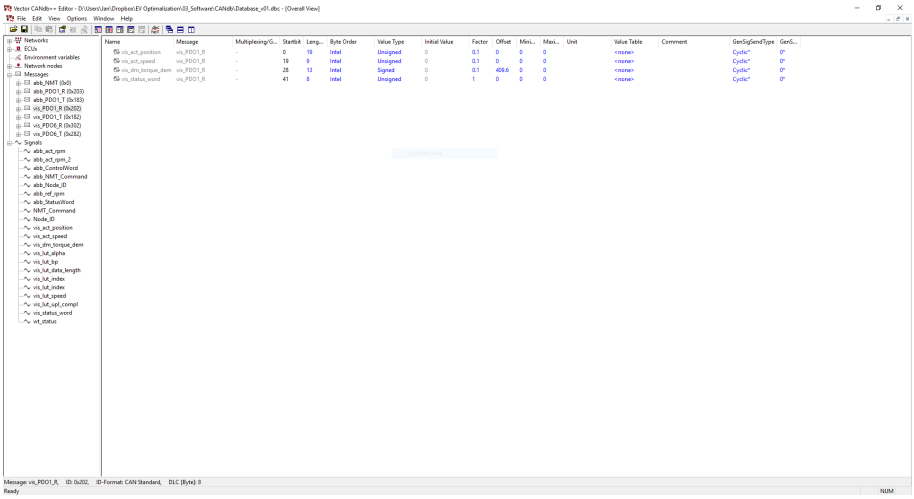


Figure 3.8: Vector CANdb++ Editor

3.3.2.1 dSpace and PIC33 communication

The RS232 communication protocol is established between the dSpace as a master and the PIC33 as a slave. The PIC33 emulates signals of the control panel of the dynamometer control system. The communication protocol is defined in table 3.8.

The master sends a message every 100ms and the slave answers. The message consists of:

- 1. byte - message length (number of bits including the length information)
- 2. byte - message type
- 3. to n. byte - data

The message is encrypted as follows: The first character is '*'=0x2A, each of the following words is divided into two nibbles in hex. The nibble characters are represented by the corresponding ASCII code in hex. The message is supplemented with a control sum of all ASCII codes except the '*', and coded in the same way as the rest of the message. The message types are defined in table 3.9.

The Control mode is 0 for manual control and 1 for remote control, the DM mode is zero for drive mode and 1 for brake mode. The Error signal does not equal 0 in an error state. The torque reference signal is multiplied by 100 to increase the precision.

Msg name	ID	Signal	Start	Length	Type	Factor	Offset
abb_NMT	0x0	NMT_Command	0	8	U	1	0
		Node_ID	8	8	U	1	0
abb_PDO1_R	0x203	abb_ControlWord	0	16	U	1	0
		abb_ref_rpm	16	16	U	1	0
abb_PDO1_T	0x183	abb_StatusWord	0	16	U	1	0
		abb_act_rpm	16	16	U	1	0
		abb_act_rpm_2	32	16	U	1	0
vis_PDO1_R	0x202	vis_act_position	0	19	U	0.1	0
		vis_act_speed	19	9	U	0.1	0
		vis_dm_torque_dem	28	13	S	0.1	409.6
		vis_status_word	41	8	U	1	0
vis_PDO1_T	0x182	wt_status	0	8	U	1	0
		vis_lut_data_length	8	13	U	1	0
vis_PDO6_R	0x302	vis_lut_index	0	16	U	1	0
vis_PDO6_T	0x282	vis_lut_bp	0	19	U	0.1	0
		vis_lut_speed	19	9	U	0.1	0
		vis_lut_alpha	28	19	S	1e-5	1.5
		vis_lut_index	44	16	U	1	0
		vis_lut_upl_compl	60	1	U	1	0

Table 3.7: List of CAN messages

Property	Value
Speed	$19200 bits/s$
Data length	$8 bits$
Parity bit	No
Number of stop bits	$1 bit$
Handshaking	No

Table 3.8: RS232 communication protocol

3.4 Control System

The control system is developed in Matlab/Simulink environment using Stateflow toolbox. It enables a quick and comprehensible development of the state machines in a graphical way. The control system is designed as shown in figure 3.9. The Main Logic reads the status of the subordinate logic blocks and sends commands to these blocks according to the status information. The subordinate logic blocks execute the commands, send their status and communicate with actuators by Interface block. Each of the *Logic* blocks consists of a state machine and a sequence. This is described in greater detail in the following sections. To make the model more readable, the signals are named according to the rule

Message type	Message composition
Synchronization message	[02][FF]
Switch to manual control	[02][01]
Switch to remote control	[02][02]
Switch to drive mode	[02][03]
Switch to brake mode	[02][04]
Set ramp	[03][05][Ramp]
Set torque reference	[04][06][T HB][T LB]
Status	[08][80][Control mode][DM mode][T HB][T LB][Ramp][Error]

Table 3.9: RS232 communication

to_from_quantity_unit, the states and the commands are defined as an enum.

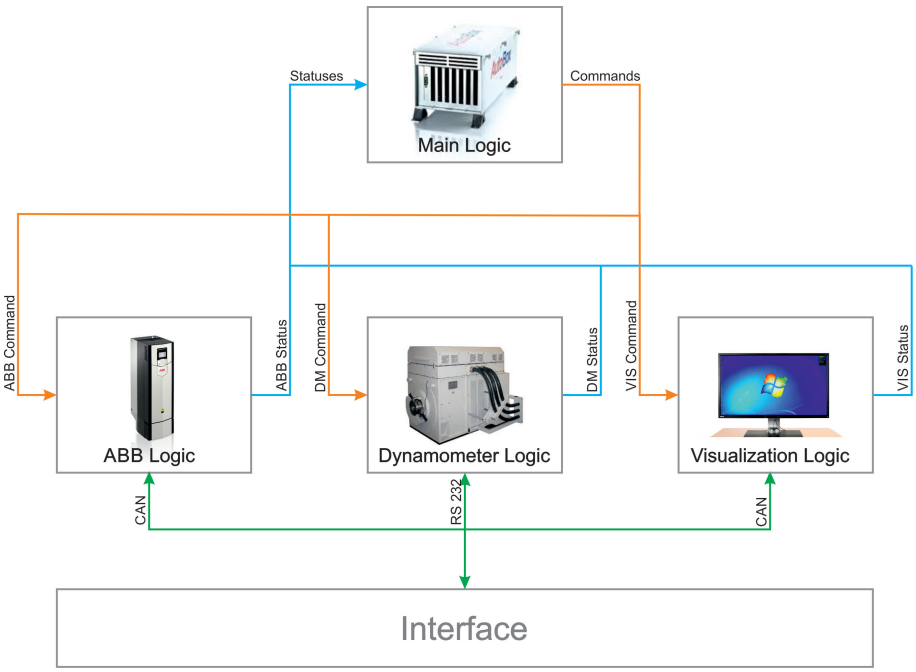


Figure 3.9: Control logic

3.4.1 Main Logic

The *Main Logic* is the top layer control logic. It reads the status of the subordinate logic blocks and sends commands to these blocks. The Main Logic state machine starts in *Init* state. In this state, the Main Logic sequence is disabled and the state machine waits until it receives *Initialized* status from all of the subordinate blocks. Then it goes to the *MeasPrepar* state. In this state, it sends command to the visualization to initialize data. When the data is initialized, it continues to *Ready* state. It waits until the *Visualization*

block sends the *VisReadyToMeasure* status. Receiving the status, the state machine goes to *Measurement* state. The Main Logic sequence is enabled and it sends commands to all subordinate blocks to start the measurement. Receiving *measurement complete* flag or *VisStopped* status, or exceeding the speed limit, the state machine goes to *Stopped* state, stopping all of the subordinate blocks. If the speed is equal to zero and the *DMStatus* equals *DMOff*, it continues to *MeasPrepar* state to enable repeating the measurement.

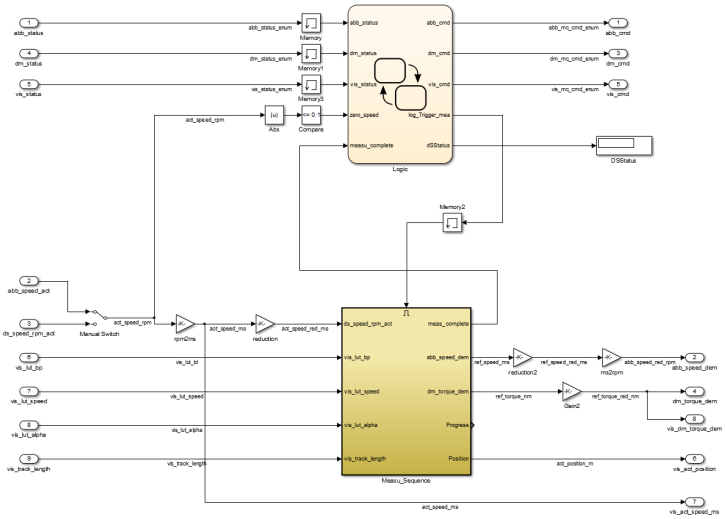


Figure 3.10: Main Logic

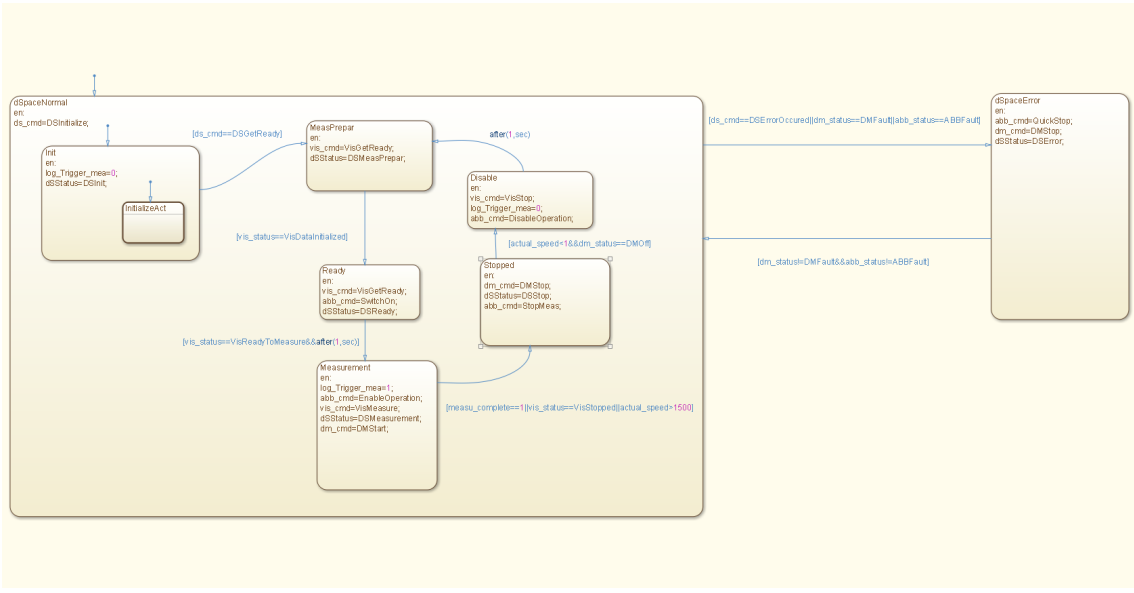


Figure 3.11: Main state mechine

When the Main Logic Sequence is enabled, it calculates the actual values of the speed demand for the ABB drive and the torque demand for the dynamometer. The actual speed

signal is integrated to get the actual position. The position signal enters look-up tables to get the values of the speed demand and the angle of inclination α at the actual position. The angle of inclination is used to calculate the drive resistance forces, leading to the load torque which is then applied by the dynamometer. The load torque is calculated as shown in equation 3.1.

$$M_L = (F_{roll} + F_{grade} + F_{air} + F_{acc})\frac{r_w}{c_{tr}} \tag{3.1}$$

Where F_{roll} is the force of the friction between the tyres and the road, F_{grade} is the gravity pull down force, F_{air} is the air resistance force, F_{acc} is the acceleration force, r_w is the radius of the wheel and c_{tr} is the transmission ratio. These forces are calculated as shown in equations 3.2, 3.3, 3.4 and 3.6.

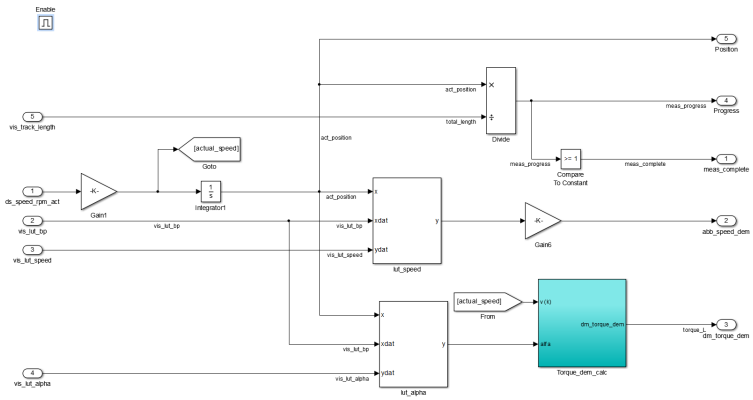


Figure 3.12: Main sequence

$$F_{roll} = mg\mu_0 cos\alpha \tag{3.2}$$

$$F_{grade} = mgsin\alpha \tag{3.3}$$

$$F_{air} = Bv^2 \tag{3.4}$$

$$F_{acc} = m\frac{dv}{dt} \tag{3.5}$$

Where m is the EV inertia, g is the gravitational constant, α is the angle of inclination, B is the air resistance constant.

As the induction motor and the power inverter do not have enough power to drive an EV, their power must be increased in a virtual way. This is done by dividing the torque and speed reference signal by a reduction constant. This leads in a virtual powertrain performance increase as shown in equation.

$$P_{virtual} = c_{red}^2 P_{real} \tag{3.6}$$

Where $P_{virtual}$ is the power of the virtual powertrain, c_{red} is the reduction constant and P_{real} is the power of the real powertrain.

3.4.2 ABB Logic

The ABB Logic is a block to handle the ABB power inverter. There are 4 tasks of the ABB logic which are necessary to be handled:

- Initialization
- Start
- Stop
- Error handling

The ABB power inverter is controlled by a *Control word* and a *Speed reference*, which are mapped to the PDO message *abb_PDO1_R*. The power inverter sends *abb_PDO1_T* with mapped *Status word* and *Actual speed*. The ABB state machine of the control system, displayed in figure 3.14, is similar to the state machine of the ABB power inverter displayed on page 99 in the datasheet of the power inverter [4].

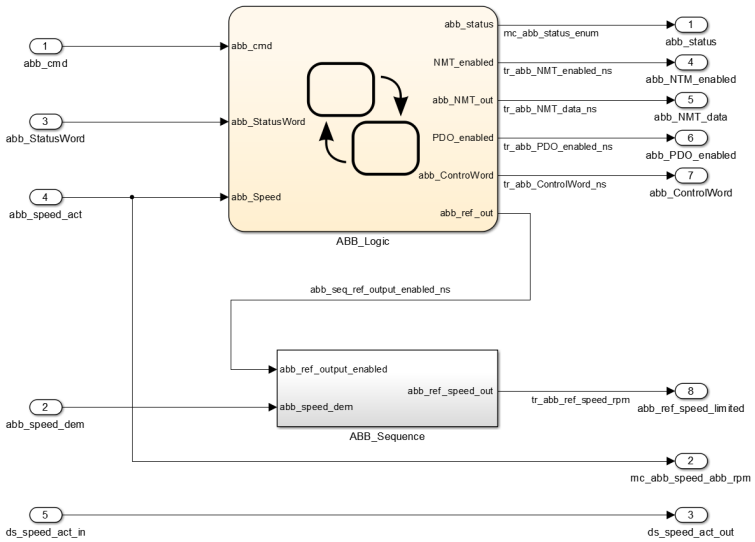


Figure 3.13: ABB Logic

Initialization *Init* is the default state of the ABB state machine. Entering this state, the NMT messages are enabled and the PDO messages are disabled, according to the CANopen protocol. The state machine automatically goes to the *Pre-Operational* state and waits for *GoToOper* command. After receiving the command, the state machine goes to *Operational* state. Entering the *Operational* state, the NMT messages are disabled,

the PDO messages are enabled, the *Control word* is set to 128 and the state machine goes to the *SwitchOnDisabled* state. After 1 second, the state machine goes to *ReadyToSwitchOn* state. Entering this state, the control word is set to 6 and the *abb_status* is set to *ABBReadyToSwitchedOn*. The initialization is done and the state machine waits until command *SwitchOn* is received.

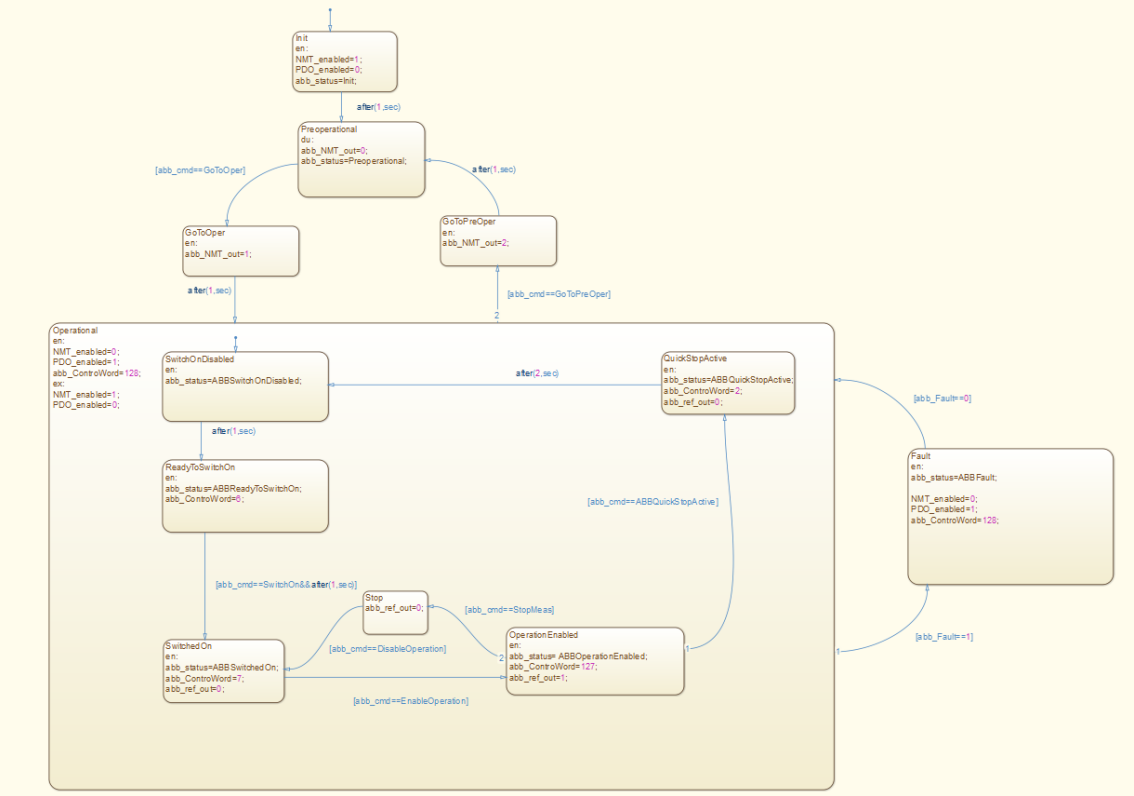


Figure 3.14: ABB state mechine

Start Receiving command *SwitchOn*,the state machine goes to state *SwitchedOn*. Entering this state, the status is set to *ABBSwitchedOn*, the *Control word* is set to 7 and the speed reference is set to 0. The state machine waits until *EnableOperation* command is received. Receiving the command, the state machine goes to *OperationEnabled* state. Entering this state, the status is set to *ABBOperationEnabled*, the control word is set to 127 and the output of the ABB Sequence block is enabled.

Stop There are two possibilities to stop the ABB power inverter operation. Receiving *StopMeas* command the state machine goes to *Stop* state. Entering this state, the speed reference is zeroed. The state machine continues to *SchitchedOn* state, if the *DisableOperation* command is received. The second option is invoked by reception of *ABBQuickStop* command, forcing the state machine to go to *QuicStopActive* state. Entering this state, the

control word is set to 2, the status is set to *ABBQuickStopActive* and the ABB Sequence output is set to zero. After 2 seconds the state machine goes to *SwitchedOnDisabled* state.

Error handling When an error is detected, the state machine goes to the *Fault* state. The status is set to *ABBFault* and the *Control word* is set to 128 to try to reset the fault.

ABB Sequence The *ABB Sequence* adjust the speed reference signal from the main logic sequence. The signal is limited by saturation block and is multiplied by *abb_ref_output_enabled* to zero the speed reference signal if the ABB state machine is in state different from *OperationEnabled* state. The sequence is displayed in figure 3.15.

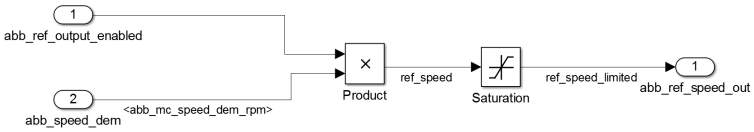


Figure 3.15: ABB sequence

3.4.3 Dynamometer Logic

The Dynamometer Logic controls the dynamometer. The dynamometer is switched on manually and then the load torque demand is controlled by RS232 message, described in section 3.3.2.1. The dynamometer state machine is displayed in figure 3.17.

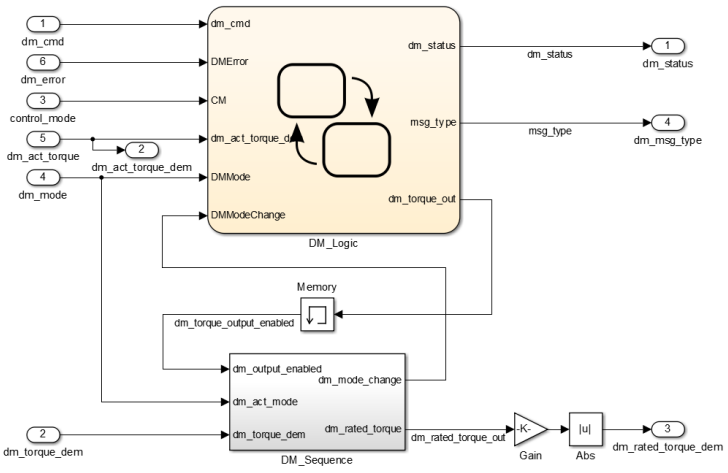


Figure 3.16: Dynamometer logic

The state machine enables the torque demand output and control communication with the dynamometer control unit. In the *DMOff* state, the torque demand output is disabled,

and the synchronization message is sent. Receiving *DMStart*, the state machine goes to *DMOn* state through the *SwitchToRC* state. The *SwitchToRC* state set the message type to *DMRemote* in order to switch the control mode to remote. When the remote control mode is submitted by the state message, the state machine continues to the *DMOn* state. Entering *DMOn* state, the torque demand output is enabled and the status is set to *DMOn*. In this state, the torque reference is sent to the control unit and the dynamometer mode starts switching according to the sign of the torque demand. Receiving *DMStop* command, the state machine goes to *DMOff* state through 2 states. The first one zeros the torque demand and the second one switches the control mode to manual. When an error state is detected, the state machine goes to *DMError* state and set the status to *DMFaul*.

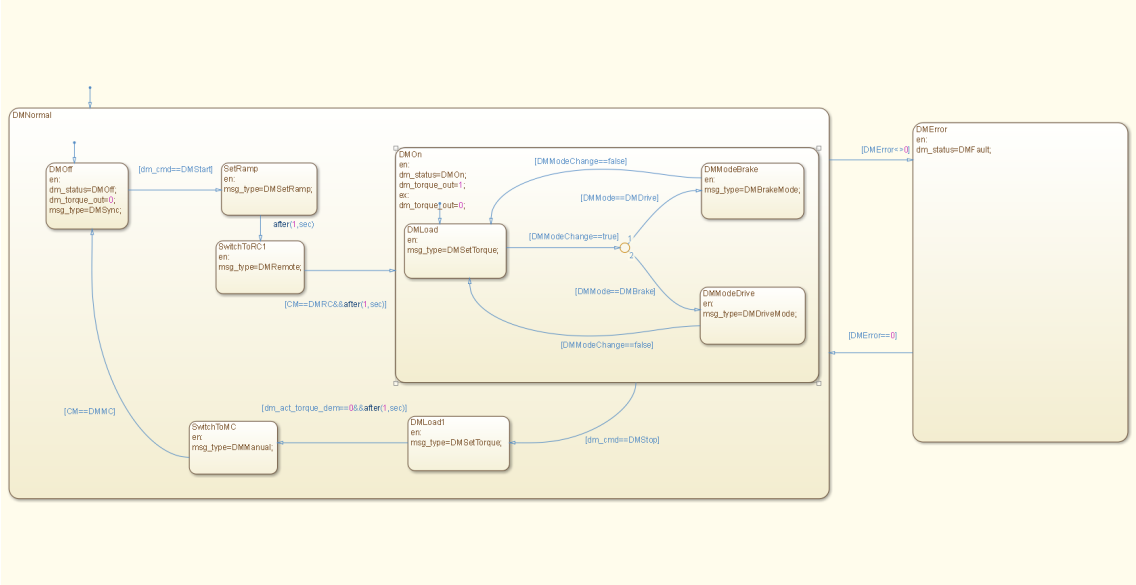


Figure 3.17: Dynamometer state mechine

The Dynamometer sequence adjusts the torque reference signal to meet the safety requirements. The torque reference signal is limited, rated and multiplied by *dm_torque_out* signal to zero the torque reference when the DM state machine is in *DMoff* state. The dynamometer sequence is displayed in figure 3.18. The lower part detects the change of the sign in the torque demand signal to change the dynamometer mode.

3.4.4 Visualization Logic

The visualization logic is a block providing communication with the user through the windows real-time target application running on the user's computer. The Visualization logic has two parts. The first is to upload the track data using *vis_PDO6_T* and *vis_PDO6_R*. The second part is a visualization of measurement progress using *vis_PDO1_R*. The windows real-time application sends its status by *vis_PDO1_T*.

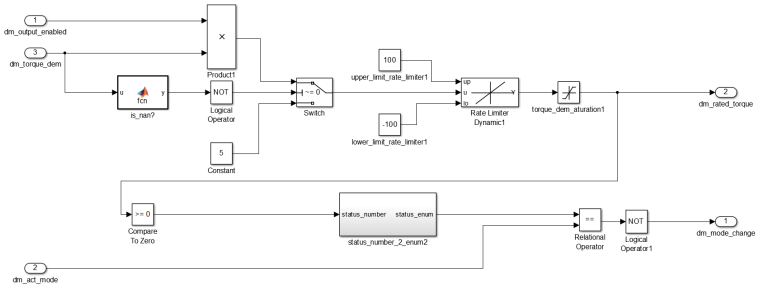


Figure 3.18: Dynamometer sequence

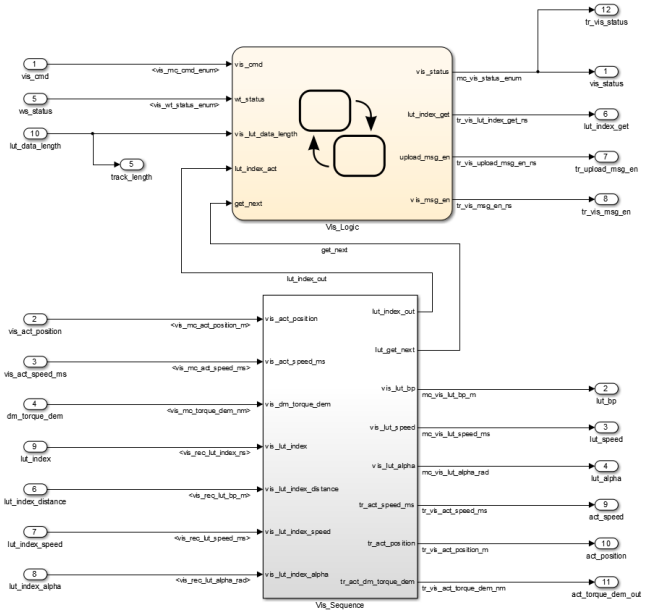


Figure 3.19: Visualization logic

The *Visualization* state machine starts in *Idle* and waits until *WTOnline* status is received from the user computer, then it goes to *Normal* state, starting in *CheckData* sub-state. The state machine disables the the visualization messages and waits for the command deciding whether to upload new data or to continue without upload.

If an upload is asked, it goes to the *UploadData* state, enabling the upload message and entering the *Upload* sub-state. In this state, the state machine controls the data upload. It asks for an index of the data table. It compares the current asked index value of the state machine with the value of index asked by *Visualization Sequence*. If the indexes are equal and the *get_next* signal is equal to 1, it goes to the *get_next* state, incrementing the asked index. However, when the asked index is equal to the data length, it goes to *Initialized* state instead of the *get_next* state. In this state, the upload status is set to complete, and the state machine continues to *DataReady* state. The upload message is

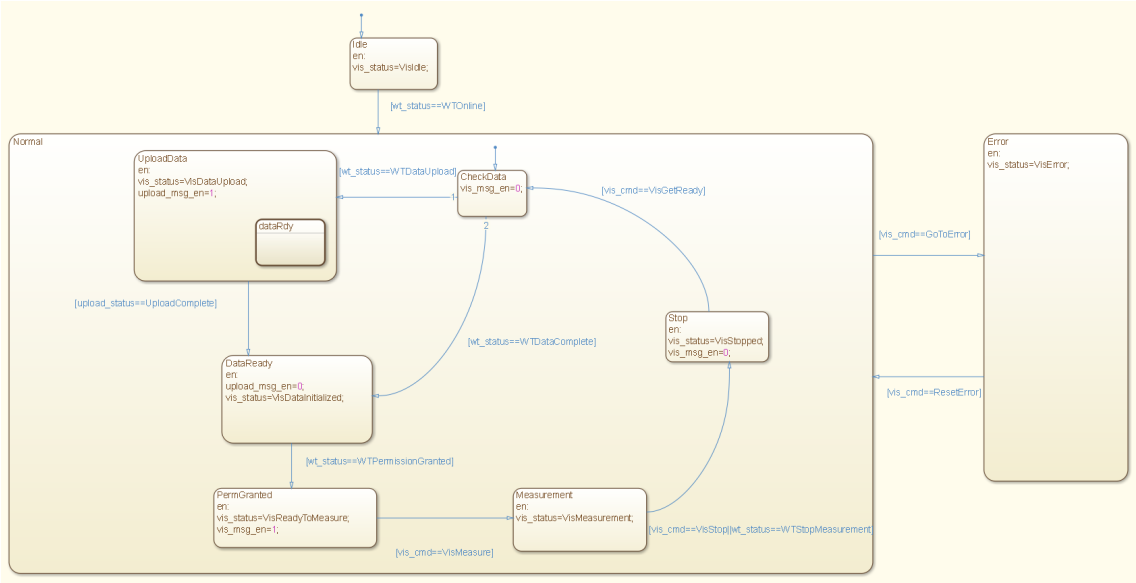


Figure 3.20: Visualization state machine

disabled and the visualization status is set to *VisDataInitialized*. It waits in this state until a permission to start measurement is granted by the user. If the permission is granted, it goes to *PermGranted* state, the visualization status is set to *VisReadyToMeasure* and the visualization message is enabled. Receiving *VisMeasure* command from the Main Logic, the state machine goes to *Measurement* state. Receiving stop command from the Main Logic or *WTStopMeasurement* status from the user computer, the state machine goes to *Stop* state, the visualization message is disabled and the status is set to *VisStopped*. It waits until *visGetReady* command is received from the Main Logic.

The visualization Sequence processes the data upload and transmits the data to be visualized. While uploading, the prepared data arrays are filled by the uploaded data at positions specified by the index. Each data array value, at position specified by the index of the uploaded data, is compared to the uploaded data to validate the proper record. If the data is valid, the next data is asked to be uploaded. The transmitted data is in the lower part of the code to be visualized.

3.4.5 Interface

The interface is divided into receive and transmit blocks. Both blocks contain blocks providing communication via CAN and RS232 communication protocols. The CAN communication is done by *RTI CAN Receive Message* and *RTI CAN Transmit Message* blocks displayed in figures 3.22 and 3.23. These blocks support loading the message specification from data file. The data file is created by CANdm++ as mentioned in section 3.3.1.2. The CAN bus parameters are specified by RTI CAN Controller Setup. The RS232 communication is provided by *DS1103SER_RX* and *DS1103SER_TX* blocks. The received and

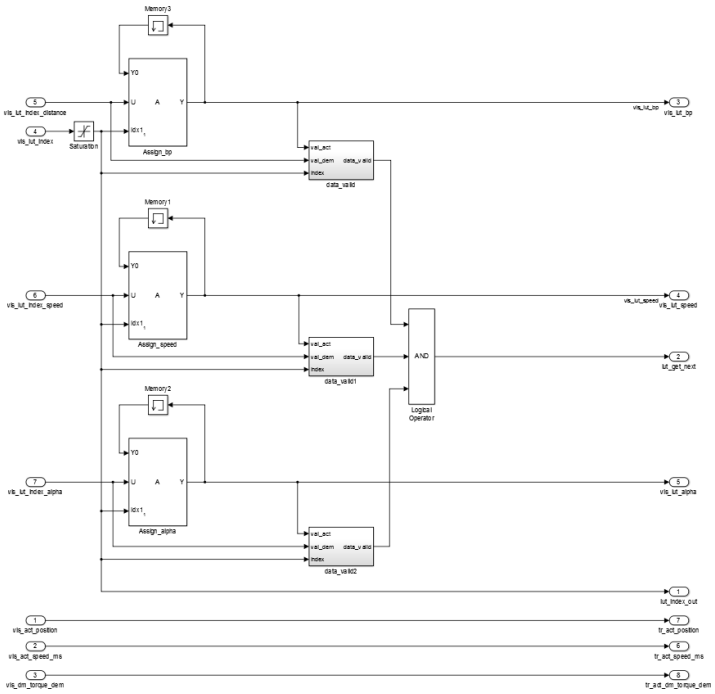


Figure 3.21: Visualization sequence

transmitted bytes are coded and decoded by S-Functions as described in section 3.3.2.1. The RS232 data reception is driven by interrupt system based on FIFO interrupt flag. The RS232 communication is specified by *Serial Setup* block according to table 3.8.

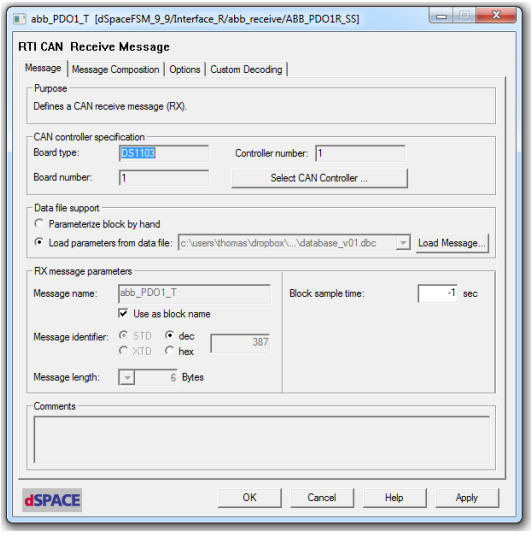


Figure 3.22: RTI CAN Receive Message

The speed incremental encoder is connected to the incremental encoder interface INC5. The speed signal is decoded by DS1103 Encoder Position block. The speed signal obtained

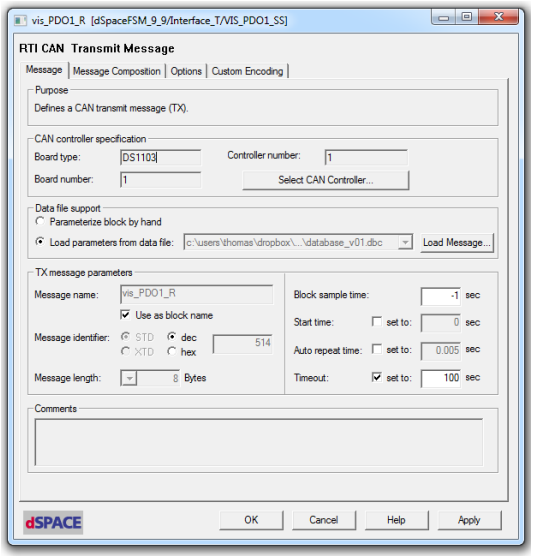


Figure 3.23: RTI CAN Transmit Message

in this way provides speed information independent of the functionality of RS232 and CAN buses.

3.4.6 Windows RT-Target

The data upload and visualization on the user's computer is provided by a real-time simulink model running in the user's computer. The real time model execution synchronization is guaranteed by the Real-Time Synchronization block from the Real-Time Windows Target library. The communication between the user's computer and the dSpace is arranged by a CAN-USB converter and the Vehicle Network Toolbox.

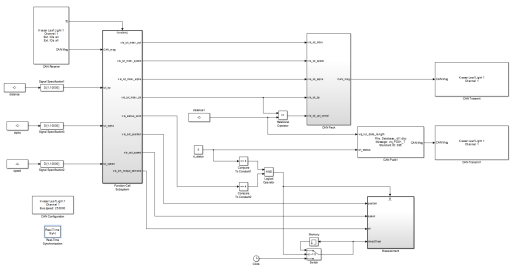


Figure 3.24: Windows RT Target model

The CAN configuration block supports the Kvaser Leaf Light device. The CAN message is received by CAN Receive block and unpacked by CAN Unpack block from the toolbox. The CAN Unpack block also supports the message specification by CAN database. The CAN message transmission is done by CAN pack and CAN Transmit blocks. In figure 3.24, the real-time simulink model is displayed. In the left part, there are arrays to be uploaded.

These arrays enter the function called subsystem. The subsystem is called by *CAN Receive* block if a message is received. The subsystem unpacks the messages *vis_PDO6_R* and the *vis_PDO1_R*. An index of the arrays to be uploaded is obtained from the *vis_PDO6_R*. The data from the *vis_PDO1_R* is visualized. The subsystem is displayed in figure 3.25. The data to be uploaded is then packed as *vis_PDO6_T* and sent. The *WTStatus* and data length are packed and sent as *vis_PDO1_T*. The Measurement subsystem helps to visualize the received data, however, it is not a part of this thesis.

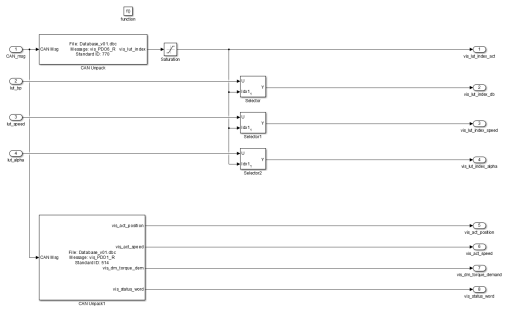


Figure 3.25: Windows RT Target message receive block

3.5 Safety features

The safety features are a necessary part of the automated simulation system. As there are two motors, the drive motor and the dynamometer, connected by a shaft and controlled independently, the main goal of the safety features is to watch the angular speed of the machine-set and to safely switch off the machine-set when an error occurs. The safety features can be divided according to the device they are implemented in. The safety features are listed in table 3.10.

The command to switch the machine-set off is done in three steps:

- Set torque reference to zero.
- Set speed reference to zero.
- Switch the DM off when the zero torque reference is reached.
- Switch the ABB off when zero speed is reached.

Designation	**Device**	**Trigger**	**Action**
Speed ref. limit	dSpace	Static	Limits the speed ref. for the ABB drive
Torque ref. limit	dSpace	Static	Limits the torque ref. for the DM drive
Torque ref. rate limit	dSpace	Static	Limits the rate of the torque ref.
Speed limit	dSpace	Exceeding	Shuts down the machine set
DM error	dSpace	DM error	Shuts down the machine set
ABB error	dSpace	ABB error	Shuts down the machine set
Speed limit	ABB	Exceeding	ABB goes to error state
Power limit	ABB	Exceeding	ABB goes to error state
Torque limit	ABB	Exceeding	ABB goes to error state
Current limit	ABB	Exceeding	ABB goes to error state
Communication loss	ABB	Loss	ABB goes to error state
Communication loss	DM	Loss	DM goes to error state
Speed limit	DM	Exceeding	DM goes to error state

Table 3.10: Safety features

Chapter 4

Results

In this chapter the results of this Master Thesis are summarized. There are 2 main results of this Master Thesis:

1. The mathematical model result
2. The simulation system result

4.1 Mathematical model result

The mathematical model of the EV powertrain is constructed in chapter 2. The components of the powertrain were modelled, identified and validated in chapter 2. The simulation results of the components are in good agreement with the measurement. The results are discussed in the following sections.

4.1.1 Induction motor

The asymmetry of the 3-phase current system is partly caused by the offset of the measured voltage. The starting curve has the same slope and rise time as the measured curve.

4.1.2 Direct torque control

The hysteresis regulation of the torque and the stator magnetic flux linkage space vector magnitude are displayed in figure 2.19. The error does not exceed markedly the bounds specified by the regulator in both cases. The speed regulation is displayed in figure 2.18. The speed transient response of the model is comparable with the transient response of the real device. The current curves of the model and the real device have almost the same amplitude, however, the duration of the transient response of the model is a little bit longer. The trajectory of the end point of the stator magnetic flux linkage space vector is round in both the simulation and the real device measurement. The shift of the trajectory of the real device is caused by the residual voltage or current offset.

4.1.3 Battery cell

The battery cell equivalent circuit parameters are functions of the SoC. The identified parameters of the battery cell are displayed in figure 2.25. The open circuit voltage is in agreement with the data sheet of the battery cell. The *Serial resistance*, the *RC TS network resistance* and the *RC TL network resistance* are almost constant and rise in the range of SoC $0\% - 30\%$. This is caused by the chemical reactions inside the battery cell. A similar trend is observed at values of the capacity of the RC network representing the long-time constant. The value of the capacity is constant in the range of SoC $100\% - 70\%$ and drops of $0\% - 30\%$. The trend of value of the capacity representing the short time constant cannot be clearly determined. This is caused by the inaccuracy of the measurement.

4.1.4 Efficiency map

The efficiency maps of the powertrain at VTP Roztoky and the powertrain model are compared in section 2.5.3. The efficiency of the powertrain model is a little bit lower in the area of small load and the area of small speed.

4.2 Simulation system result

The simulation system controls the speed of the drive motor and load torque of the dynamometer. As the torque reference rate is limited by the ramp of the control system of the dynamometer, the dynamometer does not meet the requirements for dynamics. However, the torque reference, calculated online by the model of the EV, is rated too in order to test the functionality of the simulation system. The results of the HW and SW simulations of drive along the test drive *Snezne Kratka* are displayed in figures 4.1, 4.2, 4.3 and 4.4. The simulation simulates drive of the EV along a track located at Vysočina from Sněžné to Krátká. The input of the simulation, the speed profile and the profile of angle of inclination are obtained from the application developed by Bc. Tomáš Kacetl. This application, with track specification and track altitude profile, is displayed in figure 4.5.

4.3 Summary

The energy consumption of the mathematical model in various operating states described by the efficiency map is similar to the real device. As the measurement system of the test bench is not prepared to measure the energy consumption during the drive simulation yet, it is not possible to compare total energy consumption of the HW and the SW simulations. The mechanical quantities, the speed and the torque, were measured manually during the drive. The PI speed regulator, despite the variable load torque, keeps minimal speed error during the drive. The load torque is also regulated with minimal error, however, the torque reference is limited by the ramp implemented in the dynamometer control system, so the calculated load torque reference cannot be applied yet.

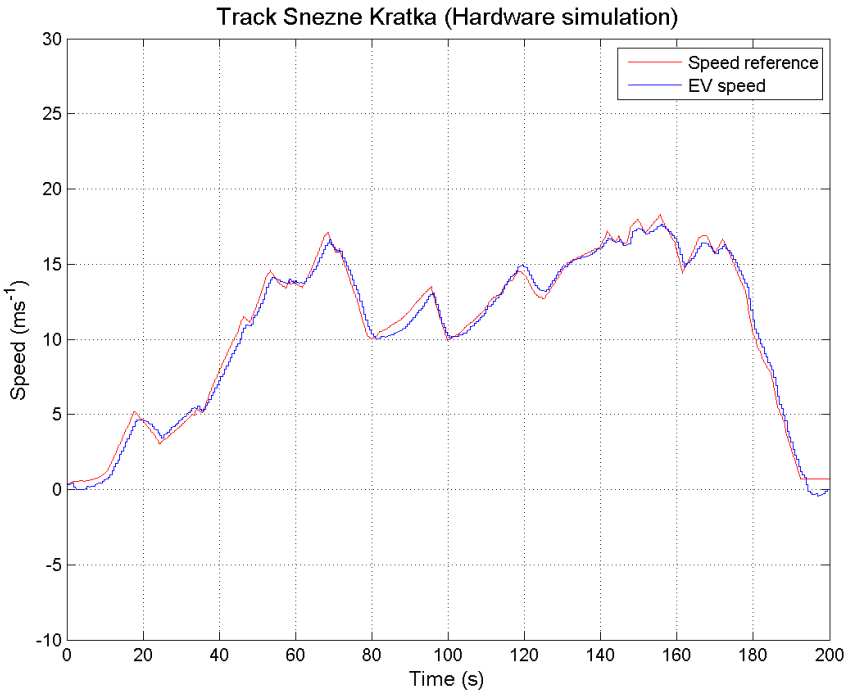


Figure 4.1: Speed control of HW simulation

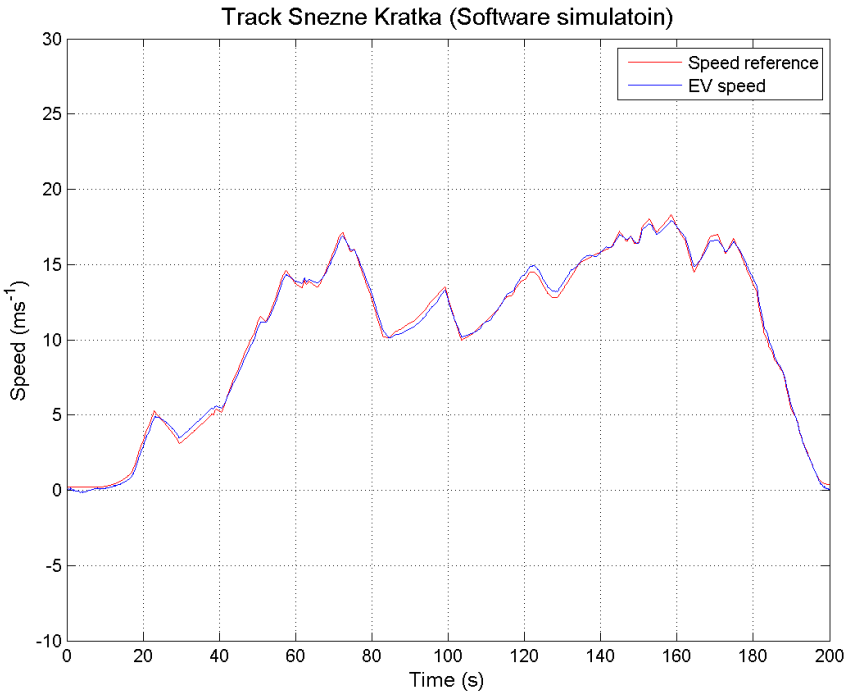


Figure 4.2: Speed control of SW simulation

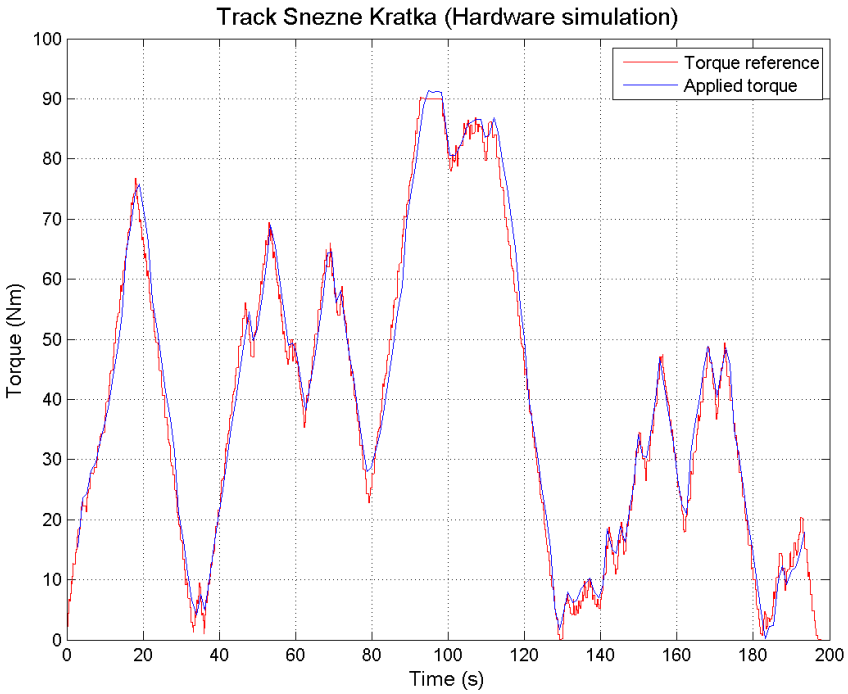


Figure 4.3: Load torque control of HW simulation

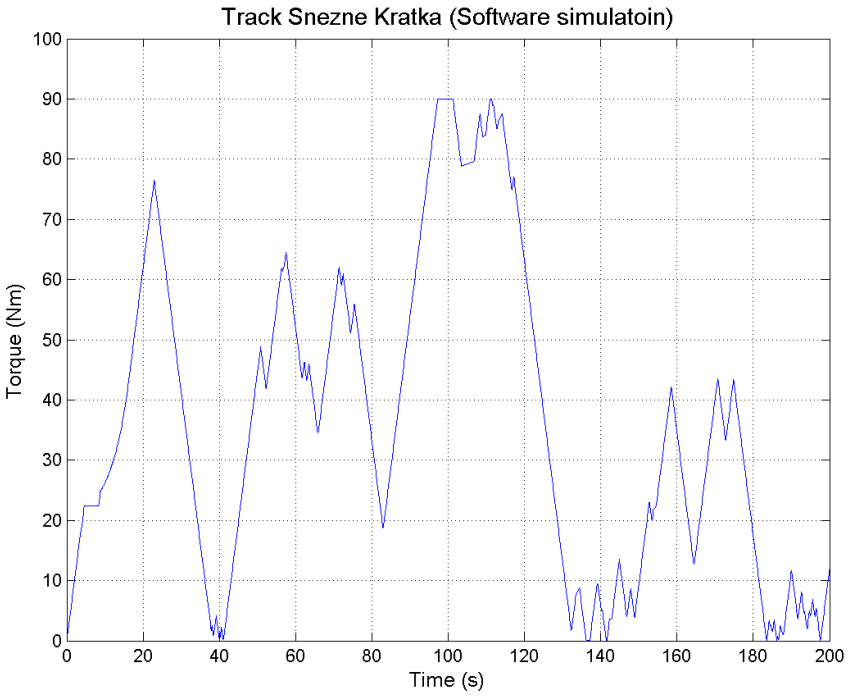


Figure 4.4: Load torque control of SW simulation

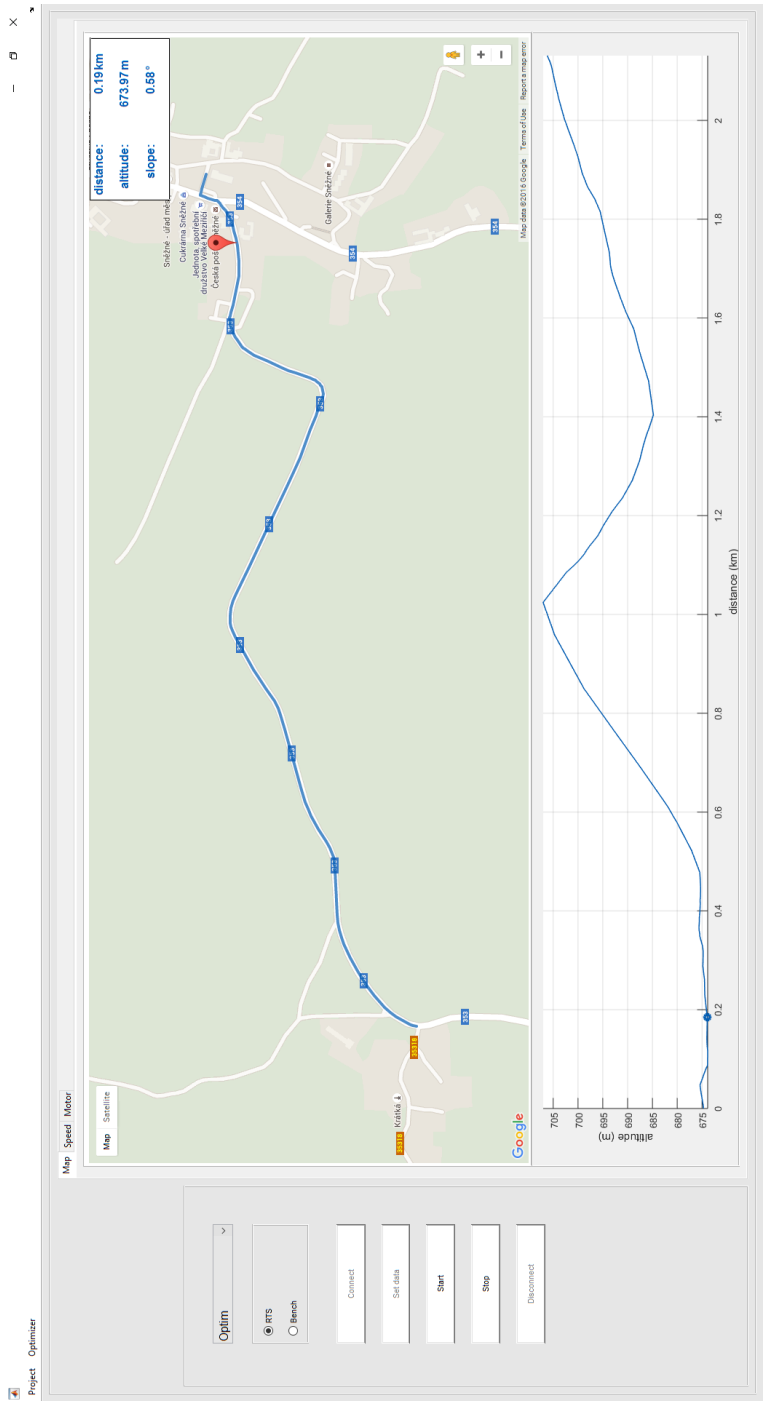


Figure 4.5: Track specification

Chapter 5

Conclusion

In the first chapter, a mathematical model of the EV powertrain is introduced. The model consists of the mathematical models of the components of the EV powertrain, based on equivalent circuit. These modes are joined together to form a mathematical model of the powertrain. This gives us an opportunity to observe the power flow in the powertrain and the behaviour of the powertrain powered by the battery pack in respect to the transient and the steady state. In the first chapter, there is also description of the system identification of the induction motor and the battery cell. The identification methods are then applied on the real devices, and the simulation results are compared with the measurement on real devices. In the second part of this thesis, the simulation system to control the test bench in VTP Roztoky is introduced. This simulation system provides a hardware simulation of EV drive along a track with a known altitude according to a predefined speed of the vehicle. During the real-time simulation of the EV drive, the drive resistance forces are calculated and applied by the controlled dynamometer as the load torque. The induction motor in the simulation system does not have enough power to simulate the drive motor of the real EV. In order to run the simulation with such an induction motor, the power of the motor is increased in a virtual way by decreasing the final torque and speed reference by the reduction constant. The rate of the torque reference is limited to $10Nm/s$. This rate limit comes from the torque reference ramp implemented in the control system of the dynamometer. The safety features are implemented in the ABB drive, the dynamometer control system and in the dSpace control algorithm. These features watch mainly the machine's set angular speed, and ensure a safe switch-off of the machine set if an error occurs.

5.1 Summary

The components of the RV powertrain in the VTP Roztoky was modelled successfully and joined together to model the whole powertrain. The efficiency of the powertrain was compared to the model in various operating states. A control system for EV drive was

developed and implemented in dSpace system. Safety features to run the test bench were coordinated and implemented into devices of the control system.

5.2 Contributions of the Master Thesis

This master thesis gives insights into modelling of the EV powertrain and discusses the methods of system identification of the components of the powertrain. The mathematical model built in this thesis is a model of a real powertrain used in the simulation system, simulating the EV drive.

In the second part the development of the control system of the test bench is described. The control system is a part of the HW simulation of the EV drive. The algorithm, which calculates the load torque reference for the dynamometer in respect to the drive resistance forces on-line, is introduced. The safety features of the control system are coordinated to ensure a save test bench run.

References

[1] REEDY, T. B. *Linden's handbook of batteries*. ISBN 978-0-07-162421-3, New York: McGraw-Hill, c2011.

[2] ABB. ACS880-01 wall-mounted industrial drives. 2016. Available from: `http://abbib.cloudapp.net/public/default/product/9AAC172656/presentation`

[3] dSPACE GmbH. Exclusive Offers for Academia 2012/2013. 10 2012.

[4] ABB. *Users manual FCAN-01 CANopen adapter module*. ABB, Affolternstrasse 44 CH-8050 Zurich Switzerland, 02 2012.

[5] Mohan, N. *Advanced electric drives: Analysis, Control, and Modeling Using MATLAB/Simulink*. ISBN 978-1-118-48548-4, John Wiley Sons, Inc., Hoboken, New Jersey, c2014.

[6] Talib, M. H. N.; Ibrahim, Z.; Rahim, N. A.; et al. Comparison Analysis of Indirect FOC Induction Motor Drive using PI, Anti-Windup and Pre Filter Schemes. *International Journal of Power Electronics and Drive Systems*, volume 5, no. 2, 10 2014: pp. 219,229.

[7] Gadoue, S. M.; Giaouris, D.; Finch, J. Tuning of PI speed controller in DTC of induction motor based on genetic algorithms and fuzzy logic schemes. *Proceedings of the 5th International Conference on Technology and Automation*, 2005: pp. 85–90.

[8] LEONHARD, W. *Control of eletrical drives*. ISBN 3-540-59380-2, Springer-Verlag Berlin Heidelberg, 1996.

[9] prof. Ing. Valeria Hrabovcova a spol. *Meranie a modelovanie elektrickych strojov*. ISBN 978-80-8070-924-2, EDIS, 2009.

[10] Jan BASTA, V. K. *Mereni na elektrickych strojich*. Praha: Statni nakladatelstvi technicke literatury, 1962.

[11] VOZENILEK, V. N. a. P. M., Petr. *Elektromechanicke menice*. ISBN 80-01-03137-3, Praha: Ceske vysoke uceni technicke, 2015.

[12] MOHAN, N. *Power electronics: a first course*. Hoboken, N.J., Wiley, 2012.

[13] Al-Refai, M. Matlab/Simulink Simulation Model for Direct Torque Control Based On Space Vector Modulation (DTC-SVM) of Induction Motor Drive. In *Proceedings of World Academy of Science, Engineering and Technology*, 76, World Academy of Science, Engineering and Technology (WASET), World Academy of Science, Engineering and Technology (WASET), 04 2013, p. 658.

[14] PAVELKA, Z. C. a. J. J., Jiri. *Elektricke pohony*. ISBN 80-01-01411-8, Praha: Ceske vysoke uceni technicke, 1996.

[15] Kroeze, R. C.; Krein, P. T. Electrical battery model for use in dynamic electric vehicle simulations. IEEE, 2008, pp. 1336–1342.

[16] Rahimi-Eichi, H.; Baronti, F.; Chow, M.-Y. Online Adaptive Parameter Identification and State-of-Charge Coestimation for Lithium-Polymer Battery Cells. *IEEE Transactions on Industrial Electronics*, volume 61, no. 4, 2014: pp. 2053–2061.

[17] Chen, M.; Rincon-Mora, G. A. Accurate electrical battery model capable of predicting runtime and I-V performance. *IEEE Transactions on Energy Conversion*, volume 21, no. 2, 2006: pp. 504–511.

[18] GmbH, S. *Magnetic Sensor MSK320*. SIKO GmbH, Weihermattenweg 2 79256 Buchenbach GERMANY.

[19] ABB. *Users Manual HTL Encoder Interface FEN-31*. ABB, Affolternstrasse 44 CH-8050 Zurich Switzerland, 06 2010.

[20] in Automation, C. CAN lower- and higher-layer protocols. 5 2016. Available from: `http://www.can-cia.org/can-knowledge/`

[21] Romnek, I. D. Co je CANopen a jak na nej. 3 2006. Available from: `http://vyvoj.hw.cz/produkty/co-je-canopen-a-jak-na-nej.html`

Appendix A

Enclosed CD

A.1 Content of the CD

The enclosed CD includes following folder structure:

- **01_Master_thesis**

  This folder contains the master thesis in a pdf file.

- **02_Matlab**

  This folder contains two folders:

  - **01_Mathematica_model**

    This folder contains the simulink model of the powertrain and the .m file with parameters of the system.

  - **02_Simulation_system**

    This folder contains the simulation system developed in Matlab/Simulink.

- **03_Results**

  This folder contains figures with the simulation results.